

**UNIVERSIDAD DE CUENCA**  
**FACULTAD DE ARQUITECTURA Y URBANISMO**  
**CARRERA DE ARQUITECTURA**

*“INTRODUCCIÓN AL DISEÑO PARAMÉTRICO: UTILIZACIÓN DE HERRAMIENTAS DIGITALES PARA LA RESOLUCIÓN DE SUPERFICIES COMPLEJAS MEDIANTE LA APLICACIÓN DE ALGORITMOS”*

**SEBASTIÁN ANDRÉS RODAS RIVERA**  
**CRISTIAN JHOVANNY BENAVIDES PADILLA**

ARQ. RAÚL ESTEBAN CORDERO GULÁ  
DIRECTOR DEL TRABAJO DE TITULACIÓN

Trabajo de titulación previo a la obtención del Título de Arquitecto

Cuenca, marzo de 2017

## RESUMEN

El presente trabajo tiene como objetivo crear una base teórica y práctica sobre el diseño paramétrico y su uso potencial para el diseño arquitectónico, especialmente para nuestro medio, en donde se evidencia la falta de conocimiento sobre este tipo de herramientas y el potencial que estas tienen para agilizar procesos tanto para la etapa de diseño como para la construcción, con este fin se ha creado un ejercicio de diseño utilizando herramientas de programación visual que permiten generar algoritmos para ser ejecutados por la computadora, este diseño se construirá posteriormente para poder determinar qué tan factible es utilizar esta tecnología en el medio.

El trabajo se ha enmarcado en una investigación de tipo exploratoria, con un diseño experimental, se emplearon varias técnicas de recolección de datos, entre las cuales resaltan principalmente el acopio de textos y demás fuentes de información secundaria, la elaboración de información primaria y la experimentación con el software puesto a prueba.

Para la creación del algoritmo que generará el diseño se ha utilizado el software de programación visual Autodesk Dynamo en conjunto con el software BIM Autodesk Revit, estos sirvieron también como base para la creación del proceso de fabricación de las piezas que conforman el diseño, también se utilizó la tecnología de corte láser.

De esta manera se pudo concluir en las ventajas y desventajas que presenta la utilización de esta tecnología en nuestro medio, al mismo tiempo que se toma un paso importante en la adopción de esta tecnología.

Palabras clave: diseño paramétrico, algoritmos generativos, arquitectura paramétrica, software CAD/CAM, Dynamo, Revit.



## ABSTRACT

The present project has as its objective to create a theoretical and practical base about parametric design and its potential use for architectural design, especially in our environment, in which its evident the lack of knowledge about this kind of tools and the potential that these has to speed up processes both for the design stage and for construction, with this in mind a design exercise was created using visual programming tools which allows to generate algorithms that will be executed by the computer, this design will later on be built to help determine how feasible it is to use this technology in our environment.

The project has been framed as an exploratory type of investigation, with an experimental design, several techniques were used for data gathering, among which mainly stand out gathering of text and other kind of secondary information, primary information elaboration and experimentation with the software put to test.

The visual programming software Autodesk Dynamo in conjunction with BIM software Autodesk Revit has been used for the creation of the algorithm that will generate the design, there programs served as well as a basis for the creation of the manufacturing process of the pieces that make up the design, laser cutting technology was used as well.

This way the advantages and disadvantages that the use of this technology represent in our environment could be concluded, at the same time that an important step towards the adoption of this technology is made.

Keywords: parametric design, generative algorithms, parametric architecture, CAD/CAM software, Dynamo, Revit.

## ÍNDICE GENERAL

RESUMEN . . . . .	2
ABSTRACT . . . . .	3
ÍNDICE GENERAL . . . . .	4
ÍNDICE DE IMÁGENES . . . . .	15
CLÁUSULA DE DERECHOS DE AUTOR . . . . .	30
CLÁUSULA DE PROPIEDAD INTELECTUAL . . . . .	31
CLÁUSULA DE DERECHOS DE AUTOR . . . . .	32
CLÁUSULA DE PROPIEDAD INTELECTUAL . . . . .	33
DEDICATORIA . . . . .	34
DEDICATORIA . . . . .	35
AGRADECIMIENTOS . . . . .	36
PREFACIO . . . . .	37

## CAPÍTULO I 39

<b>1. ANTECEDENTES Y MARCO TEÓRICO . . . . .</b>	<b>39</b>
1.1. Aspectos generales . . . . .	39
1.1.1. Planteamiento del problema . . . . .	39
1.1.2. Objetivo de la investigación . . . . .	39

1.1.2.a.	Objetivo general . . . . .	39
1.1.2.b.	Objetivos específicos . . . . .	39
1.1.3.	Justificación . . . . .	39
1.1.4.	Preguntas de investigación . . . . .	40
1.1.5.	Hipótesis. . . . .	40
1.2.	Estado del arte . . . . .	40
1.2.1.	Antecedentes. . . . .	40
1.3.	Marco teórico . . . . .	42
1.3.1.	Diseño paramétrico. . . . .	42
1.3.1.a.	Precursores filosóficos del parametricismo . . . . .	42
	Antoni Gaudí y la Sagrada Familia. . . . .	42
	Frei Otto y el proceso de búsqueda de formas . . . . .	43
	Greg Lynn y la Blobitectura . . . . .	45
1.3.1.b.	Definición de parámetro . . . . .	45
1.3.1.c.	El Parametricismo como estilo arquitectónico . . . . .	46
1.3.1.d.	Software y diseño paramétrico . . . . .	50
1.3.1.e.	Software Autodesk Revit. . . . .	51
1.3.1.f.	Software Autodesk Dynamo . . . . .	53
1.3.1.g.	Software Rhinoceros . . . . .	53
1.3.1.h.	Software Grasshopper . . . . .	54
1.3.1.i.	Software y Arquitectura . . . . .	54

1.3.2.	Algoritmos y algoritmos generativos . . . . .	54
1.3.2.a.	Triangulación de Delaunay . . . . .	55
	Convex Hull (Casco convexo) . . . . .	56
	Algoritmo Divide and Conquer (Divide y Vencerás) . . . . .	56
	Aplicaciones de la triangulación de Delaunay . . . . .	57
1.3.2.b.	Diagrama de Voronoi. . . . .	57
	Definiciones y propiedades elementales . . . . .	57
	Algoritmo Incremental . . . . .	58
	Algoritmo Divide and Conquer (Divide y Vencerás) . . . . .	58
	Algoritmo de Fortune . . . . .	58
1.3.2.c.	Gramática de la forma (Shape Grammar) . . . . .	59
1.3.3.	Programación visual y su anatomía . . . . .	60
	Instrucciones textuales . . . . .	60
	Programación . . . . .	61
1.3.3.a.	Definición de la programación visual. . . . .	61
1.3.3.b.	Anatomía de la programación visual . . . . .	62
	Anatomía de un nodo . . . . .	62
	Puertos . . . . .	63
	Estado de los nodos . . . . .	63
	Cableado . . . . .	64
	La Biblioteca de Dynamo . . . . .	64

1.3.3.c. Componentes de la programación. . . . .	65
Datos . . . . .	65
Matemática . . . . .	65
Lógica . . . . .	65
Cuerdas . . . . .	66
Color . . . . .	66
1.3.3.d. Geometría para el diseño computacional . . . . .	66
Vectores, planos y sistemas de coordenadas . . . . .	67
Puntos. . . . .	67
Curvas . . . . .	67
Superficies . . . . .	68
Sólidos . . . . .	68
Mallas. . . . .	69
1.3.4. El CAD y CAM en el diseño y fabricación arquitectónica . . . .	70
1.3.4.a. Aplicaciones del sistema CAD/CAM . . . . .	72
1.3.4.b. Arquitectura y el sistema CAD/CAM. . . . .	72
1.3.4.c. El sistema del software BIM . . . . .	73
1.3.5. Programas seleccionados para realizar el proyecto . . . . .	74
1.3.5.a. Software Revit. . . . .	74
Ventajas. . . . .	74
Desventajas . . . . .	75

1.3.5.b. Software Rhinoceros . . . . .	75
Ventajas . . . . .	75
Desventajas . . . . .	75
1.3.5.c. Software Dynamo . . . . .	76
Ventajas . . . . .	76
Desventajas . . . . .	76
1.3.5.d. Software Grasshopper . . . . .	76
Ventajas . . . . .	76
Desventajas . . . . .	76
1.3.5.e. Valoración de los softwares BIM . . . . .	76
1.3.5.f. Valoración de los softwares paramétricos . . . . .	77
1.4. Casos de estudio . . . . .	77
1.4.1. Proyecto DIGFABMTY 1.0 . . . . .	77
1.4.2. Levantamiento del centro de la ciudad de Málaga . . . . .	78
1.5. Marco metodológico . . . . .	79
1.5.1. Tipo y diseño de la investigación . . . . .	79
1.5.2. Población y muestra para la investigación. . . . .	79
1.5.3. Técnicas e instrumentos de recolección de datos . . . . .	80
1.5.4. Técnicas de procesamiento y análisis de datos . . . . .	80
1.5.5. Actividades realizadas . . . . .	80

<b>2.</b>	<b>DISEÑO DE UN ESPACIO MEDIANTE EL USO DE ALGORITMOS GENERATIVOS . . .</b>	<b>83</b>
2.1.	Definición y abstracción de la idea, conceptualización del diseño . . . .	83
2.1.1.	Definición y abstracción de la idea . . . . .	83
2.1.1.a.	Estrategias de diseño . . . . .	83
2.1.1.b.	Idea general . . . . .	84
2.1.1.c.	Abstracción de la idea . . . . .	85
2.2.	Establecimiento de condiciones geométricas y matemáticas . . . . .	87
2.2.1.	Espacio contenedor del diseño . . . . .	87
2.2.2.	Secciones del espacio contenedor . . . . .	87
2.2.3.	Modificación de las secciones definidas . . . . .	88
2.2.3.a.	Definición de la ecuación de la onda . . . . .	89
2.2.3.b.	Traslación de las ondas a las secciones . . . . .	90
2.2.4.	Creación de la superficie del diseño . . . . .	90
2.2.5.	Triangulación de la superficie . . . . .	91
2.2.6.	Definición de los paneles . . . . .	91
2.2.7.	Colocación de los paneles . . . . .	92
2.2.8.	Softwares utilizados para el proyecto . . . . .	92
2.2.8.a.	Autodesk Revit . . . . .	92

2.2.8.b. Autodesk Dynamo . . . . .	93
2.3. Definición del algoritmo generativo para el diseño . . . . .	94
Grupo de nodos modificadores. . . . .	94
Grupo de nodos de procesos . . . . .	94
Grupo de nodos de resultados . . . . .	94
Grupo de nodos condicionales . . . . .	94
2.3.1. Instrucciones para Dynamo . . . . .	94
2.3.1.a. Definición de la base del volumen . . . . .	95
2.3.1.b. Definición de la cubierta y caras laterales del volumen . .	98
2.3.1.c. Definición de las secciones de la superficie. . . . .	100
2.3.1.d. Unión de listas. . . . .	103
Condicional lateral AB . . . . .	104
Condicional lateral CD . . . . .	106
2.3.1.e. Secciones finales. . . . .	107
2.3.1.f. Definición de laterales AD y BC . . . . .	111
Condicional para laterales AD y BC . . . . .	112
2.3.1.g. Verificación y corrección de la dirección de los vectores normales. . . . .	112
2.3.1.h. Ecuación de la propagación de la onda . . . . .	115
2.3.1.i. Definición de la superficie . . . . .	117



2.3.1.j. Triangulación de Delaunay en la superficie y exportación de elementos a Revit . . . . .	118
Determinación del Módulo A . . . . .	123
Determinación del Panel A. . . . .	124
2.4. Representación y exploración de resultados. . . . .	125
2.4.1. Modificadores de la base del volumen . . . . .	126
2.4.2. Modificadores de la cubierta del volumen. . . . .	126
2.4.3. Modificadores de las secciones de la superficie . . . . .	126
2.4.4. Modificadores para la generación de la superficie. . . . .	127
2.4.5. Modificadores de la ecuación de la propagación de la onda . . .	128
2.4.6. Modificadores para la división de la superficie . . . . .	128

## CAPÍTULO III

**133**

<b>3. ANÁLISIS DEL DISEÑO Y ESTRATEGIAS PARA SU FABRICACIÓN . . . . .</b>	<b>133</b>
3.1. Establecimiento del proceso de fabricación óptimo para el diseño . . .	133
3.1.1. Maquinaria CNC . . . . .	133
3.1.1.a. Tipos de maquinaria CNC . . . . .	133
Torno CNC . . . . .	133
Taladradora CNC . . . . .	134
Fresadora CNC . . . . .	134

Centro de mecanizado CNC . . . . .	134
Cortadora láser CNC . . . . .	134
Cortadora por chorro de agua CNC . . . . .	135
3.1.2. Algoritmos de fabricación . . . . .	135
3.1.2.a. Aplicación del algoritmo generativo . . . . .	136
3.1.2.b. Filtración de triángulos problemáticos . . . . .	138
3.1.2.c. Definición de elementos auxiliares A, B, C . . . . .	139
3.1.2.d. Codificación de elementos Módulo, Panel y Auxiliares . .	140
3.1.2.e. Definición de piezas de sujeción . . . . .	141
3.1.2.f. Codificación de piezas de sujeción . . . . .	145
3.1.2.g. Generación de aberturas en paneles. . . . .	146
3.1.2.h. Definición de paneles finales . . . . .	146
3.1.2.i. Codificación de paneles . . . . .	148
3.2. Análisis y elección de materiales . . . . .	148
3.2.1. Tipos de corte y los materiales . . . . .	148
3.2.1.a. Corte láser por vaporización . . . . .	148
3.2.1.b. Corte láser por fundido y soplado . . . . .	149
3.2.1.c. Corte láser por estrés térmico . . . . .	149
3.2.1.d. Corte láser reactivo . . . . .	149
3.2.2. Materiales . . . . .	149
3.2.2.a. Plásticos . . . . .	149

	ABS - Acrilonitrilo Butadieno Estireno . . . . .	150
	PMMA - Acrílico o Polimetilmetacrilato . . . . .	150
	PP - Polipropileno . . . . .	150
	Poliestireno . . . . .	150
3.2.2.b.	Láminas metálicas . . . . .	150
	Acero inoxidable . . . . .	150
	Acero suave . . . . .	151
	Aluminio . . . . .	151
3.2.2.c.	Espuma . . . . .	151
	XPS – Espuma de Poliestireno Extruido . . . . .	151
	Espuma de Polietileno . . . . .	151
3.2.2.d.	Otros . . . . .	152
	Tejidos . . . . .	152
	Papeles . . . . .	152
	Gomas . . . . .	152
	Maderas . . . . .	152
3.2.3.	Materiales seleccionados . . . . .	153
3.3.	Elaboración de un prototipo a escala . . . . .	153
3.3.1.	Definición de la muestra . . . . .	153
3.3.2.	Verificación de las piezas para el ensamblaje . . . . .	154
3.3.3.	Ensamblaje del prototipo . . . . .	154

3.3.4. Proceso de elaboración de la maqueta. . . . .	155
3.3.4.a. Cortado de tableros de MDF . . . . .	155
3.3.4.b. Cortado de las piezas de la maqueta a láser. . . . .	155
3.3.4.c. Recubrimiento de las piezas con adhesivo plástico . . . .	155
3.3.4.d. Armado de prismas con los paneles de MDF . . . . .	155
3.3.4.e. Colocación de elementos de sujeción . . . . .	157
3.3.4.f. Ensamblaje de una muestra a escala 1:1 . . . . .	157
3.3.5. Observaciones . . . . .	157

## **CAPÍTULO IV** **161**

<b>4. CONCLUSIONES Y RECOMENDACIONES . . . . .</b>	<b>161</b>
4.1. Conclusiones. . . . .	161
4.2. Recomendaciones . . . . .	161
BIBLIOGRAFÍA Y REFERENCIAS. . . . .	163

## ÍNDICE DE IMÁGENES

<b>CAPÍTULO I</b>	<b>39</b>
<b>Imagen 1.1.</b>	Estructura paramétrica - Imagen digital (Espacio Arquitectura, 2013) . . . . . 40
<b>Imagen 1.2.</b>	Superficies deformadas - Imagen digital (Parametric Wood, 2011) . . . . . 41
<b>Imagen 1.3.</b>	T-splines en Dynamo - Imagen digital (Kron, 2016) . . . . . 42
<b>Imagen 1.4.</b>	Modelo de cuerdas colgadas con pesos de Gaudí - Fotografía (Architizer, 2016) . . . . . 43
<b>Imagen 1.5.</b>	Experimentación de superficies mediante el uso de pompas de jabon, Frei Otto - Fotografía (Plataforma Arquitectura, 2015) . . . . . 43
<b>Imagen 1.6.</b>	Pavellón Blobwall de Greg Lynn - Fotografía (Revista PLOT, 2012) . . . . . 44
<b>Imagen 1.7.</b>	Proceso de construcción del Heydar Aliyev Center - Fotografía (Adelto, 2013). . . . . 45
<b>Imagen 1.8.</b>	Heydar Aliyev Center, representante del deconstructivismo - Fotografía (Deconstructivismo, 2015) . . . . . 46
<b>Imagen 1.9.</b>	Museo de Arte de Tel Aviv, representante del folding - Fotografía (A-cero Blog, 2015) . . . . . 46
<b>Imagen 1.10.</b>	Guangzhou Opera House, exterior - Fotografía (Equipo Arquidesign, 2011) . . . . . 47

<b>Imagen 1.11.</b>	Guangzhou Opera House, interior - Fotografía (LC&LP, 2016)	47
<b>Imagen 1.12.</b>	JS Bach Chamber Music Hall, exterior - Fotografía (DailyTonic, 2012).	47
<b>Imagen 1.13.</b>	JS Bach Chamber Music Hall, interior - Fotografía (Aminta Online, 2010)	47
<b>Imagen 1.14.</b>	Fabricación y ensamblaje del diseño paramétrico - Fotografía (Parametric Fabrication, 2011).	48
<b>Imagen 1.15.</b>	Esquema de flujo vs. algoritmo generado en Grasshopper - Imagen digital (GeometricMind, 2011).	49
<b>Imagen 1.16.</b>	Entorno de trabajo de Rhinoceros - Captura de pantalla (Simply Rhino Rhino3D Channel, 2012)	50
<b>Imagen 1.17.</b>	Entorno de trabajo de Grasshopper para Rhino - Captura de pantalla (MBWAREZ, 2011)	50
<b>Imagen 1.18.</b>	Entorno de trabajo de Autodesk Revit - Captura de pantalla (Autodesk, 2016)	52
<b>Imagen 1.19.</b>	Entorno de trabajo de Autodesk Dynamo - Captura de pantalla (Dynamo Primer, 2015)	52
<b>Imagen 1.20.</b>	Cumplimiento del criterio para la creación de la triangulación de Delaunay - Imagen digital (Peterson, 1998)	56
<b>Imagen 1.21.</b>	Triangulación de Delaunay para un conjunto de 50 puntos aleatorios - Imagen digital (Wikimedia Commons, 2007)	57
<b>Imagen 1.22.</b>	Diagrama de Voronoi para el conjunto de puntos resaltados en rojo - Imagen digital (Wiens, 2013)	58
<b>Imagen 1.23.</b>	Diagrama de Voronoi y su contraparte la triangulación de Delaunay para un mismo conjunto de puntos - Imagen digital	

	(Soriano, 2013) . . . . .	59
<b>Imagen 1.24.</b>	Aplicación de un conjunto de reglas para definir una volumetría - Imagen digital (Tan, 2014) . . . . .	60
<b>Imagen 1.25.</b>	Instrucciones gráficas, algoritmo para crear una grulla de origami - Imagen digital (Dynamo Primer, 2015) . . . . .	61
<b>Imagen 1.26.</b>	Programación visual, algoritmo para generar una circunferencia con base en un punto - Imagen digital (Dynamo Primer, 2015) . . . . .	62
<b>Imagen 1.27.</b>	Resultado del algoritmo definido anteriormente - Imagen digital (Dynamo Primer, 2015) . . . . .	62
<b>Imagen 1.28.</b>	Anatomía de un nodo - Imagen digital (Dynamo Primer, 2015)	63
<b>Imagen 1.29.</b>	Anatomía de un puerto - Imagen digital (Dynamo Primer, 2015) . . . . .	63
<b>Imagen 1.30.</b>	Estado de los nodos en Autodesk Dynamo - Imagen digital (Dynamo Primer, 2015) . . . . .	64
<b>Imagen 1.31.</b>	Cableado y flujo de trabajo en Autodesk Dynamo - Imagen digital (Dynamo Primer, 2015) . . . . .	64
<b>Imagen 1.32.</b>	Organización jerárquica de la geometría - Imagen digital (Dynamo Primer, 2015) . . . . .	65
<b>Imagen 1.33.</b>	Cuadro explicativo de la jerarquía de los datos geométricos - Imagen digital (Dynamo Primer, 2015) . . . . .	66
<b>Imagen 1.34.</b>	Vector dentro de un sistema de coordenadas tri-dimensional - Imagen digital (Dynamo Primer, 2015) . . . . .	67
<b>Imagen 1.35.</b>	Representación de un plano en el espacio - Imagen digital	

	(Dynamo Primer, 2015) . . . . .	67
<b>Imagen 1.36.</b>	Representación del origen de un sistema de coordenadas X, Y, Z - Imagen digital (Dynamo Primer, 2015) . . . . .	67
<b>Imagen 1.37.</b>	Representación de un punto en el espacio - Imagen digital (Dynamo Primer, 2015) . . . . .	68
<b>Imagen 1.38.</b>	Representación de una línea y su proyección hacia el infinito - Imagen digital (Dynamo Primer, 2015) . . . . .	68
<b>Imagen 1.39.</b>	Representación de una polilínea (curva abierta) y un polígono (curva cerrada) - Imagen digital (Dynamo Primer, 2015) . . . . .	68
<b>Imagen 1.40.</b>	Representación de una curva NURBS - Imagen digital (Dynamo Primer, 2015) . . . . .	69
<b>Imagen 1.41.</b>	Superficie y representación de un punto según sus coordenadas U, V - Imagen digital (Dynamo Primer, 2015) . . . . .	69
<b>Imagen 1.42.</b>	Superficies NURBS representadas como una grilla de curvas NURBS - Imagen digital (Dynamo Primer, 2015). . . . .	69
<b>Imagen 1.43.</b>	Representación de una polisuperficie - Imagen digital (Dynamo Primer, 2015) . . . . .	69
<b>Imagen 1.44.</b>	Representación de dos tipos de sólidos diferentes - Imagen digital (Dynamo Primer, 2015). . . . .	69
<b>Imagen 1.45.</b>	Diferentes tipos de operaciones booleanas en sólidos - Imagen digital (Dynamo Primer, 2015). . . . .	69
<b>Imagen 1.46.</b>	Diferencia entre influencia global e influencia local (NURBS vs mallas) - Imagen digital (Dynamo Primer, 2015) . . . . .	69
<b>Imagen 1.47.</b>	Entorno de trabajo de un programa CAM y simulación de un conjunto de instrucciones para fabricación - Captura de	



	pantalla (BobCAD-CAM, 2014) . . . . .	70
<b>Imagen 1.48.</b>	Entorno de trabajo de un programa CAD, diseño de piezas para fabricación - Captura de pantalla (SolidSmack, 2009) . .	70
<b>Imagen 1.49.</b>	Maquinaria CNC de 5 ejes, permite gran flexibilidad al crear partes complejas - Fotografía (SHL GROUP, 2016). . . . .	71
<b>Imagen 1.50.</b>	Maquinaria CNC de tipo Gantry, utilizada para la industria - Fotografía (SHL GROUP, 2016) . . . . .	71
<b>Imagen 1.51.</b>	Softwares y formatos BIM más utilizados en el medio - Imagen digital (Fabricantes BIM, 2016) . . . . .	73
<b>Imagen 1.52.</b>	Tabla de valoración para los softwares BIM - Imagen digital (Elaboración propia, 2016). . . . .	77
<b>Imagen 1.53.</b>	Tabla de valoración para los softwares paramétricos - Imagen digital (Elaboración propia, 2016) . . . . .	77
<b>Imagen 1.54.</b>	Proceso de doblado del módulo básico que conforma el pabellón - Imagen digital (ArchDaily, 2015) . . . . .	78
<b>Imagen 1.55.</b>	Resultado final del ensamblaje del pabellón - Fotografía (ArchDaily, 2015) . . . . .	78
<b>Imagen 1.56.</b>	Algoritmo utilizado para obtener el contorno de las edificaciones del centro de la ciudad de Málaga - Captura de pantalla (GARQUITECTOS, 2016). . . . .	79
<b>Imagen 1.57.</b>	Resultado de la aplicación del programa visual dentro del entorno de trabajo de Revit - Captura de pantalla (GARQUITECTOS, 2016) . . . . .	79

<b>Imagen 2.1.</b>	Propagación de ondas en la superficie del agua luego de haber recibido un estímulo - Fotografía (Averill & Eldredge, 2016) .	84
<b>Imagen 2.2.</b>	Representación de la propagación de ondas en la superficie del agua - Imagen digital (Wikipedia, 2016) . . . . .	85
<b>Imagen 2.3.</b>	Molécula de agua, diferentes representaciones - Imagen digital (Bonsai Menorca, 2015) . . . . .	85
<b>Imagen 2.4.</b>	Estructura tetraédrica de la molécula de agua - Imagen digital (Ciencia y Salud, 2012) . . . . .	86
<b>Imagen 2.5.</b>	Enlaces entre moléculas de agua, enlaces de hidrógeno - Imagen digital (QuimiTube.com, s.f.) . . . . .	86
<b>Imagen 2.6.</b>	Alturas de un triángulo y su ortocentro, punto H - Imagen digital (Elaboración propia, 2016) . . . . .	87
<b>Imagen 2.7.</b>	Medianas de un triángulo y su baricentro, punto G - Imagen digital (Elaboración propia, 2016) . . . . .	87
<b>Imagen 2.8.</b>	Base y cubierta del espacio contenedor del diseño - Captura de pantalla (Elaboración propia, 2016) . . . . .	88
<b>Imagen 2.9.</b>	Secciones creadas en el espacio contenedor del diseño - Captura de pantalla (Elaboración propia, 2016). . . . .	88
<b>Imagen 2.10.</b>	División de las secciones y creación de puntos de control - Captura de pantalla (Elaboración propia, 2016). . . . .	89
<b>Imagen 2.11.</b>	Desplazamiento de los puntos de control, definición de curvas NURBS - Captura de pantalla (Elaboración propia, 2016) . .	89

<b>Imagen 2.12.</b>	Gráficos de la ecuación de la onda, en función del tiempo $t$ y el espacio $x$ - Captura de pantalla (Elaboración propia, 2016).	90
<b>Imagen 2.13.</b>	Creación de $n$ número de puntos a lo largo de las secciones NURBS - Captura de pantalla (Elaboración propia, 2016)	90
<b>Imagen 2.14.</b>	Secciones definitivas, mediante el uso de curvas NURBS - Captura de pantalla (Elaboración propia, 2016).	91
<b>Imagen 2.15.</b>	Superficie creada mediante la interpolación de las curvas en las secciones - Captura de pantalla (Elaboración propia, 2016)	91
<b>Imagen 2.16.</b>	Triangulación de Delaunay aplicada a la superficie del diseño - Captura de pantalla (Elaboración propia, 2016)	92
<b>Imagen 2.17.</b>	Representación básica de los paneles a ser colocados - Captura de pantalla (Elaboración propia, 2016)	92
<b>Imagen 2.18.</b>	Codificado por color según la perpendicularidad de la cara en relación al vector definido por la incidencia solar - Captura de pantalla (Elaboración propia, 2016)	93
<b>Imagen 2.19.</b>	Abertura generada en los paneles, valores 0, 0.2, 0.4, 0.6, 0.8, 1 - Captura de pantalla (Elaboración propia, 2016).	93
<b>Imagen 2.20.</b>	Menú para crear un nuevo proyecto - Captura de pantalla (Elaboración propia, 2016).	94
<b>Imagen 2.21.</b>	Algoritmo que define la base del volumen y sus modificadores - Captura de pantalla (Elaboración propia, 2016)	95
<b>Imagen 2.22.</b>	Algoritmo que define la cubierta del volumen y sus modificadores - Captura de pantalla (Elaboración propia, 2016)	97

<b>Imagen 2.23.</b>	Modificadores y resultados del grupo que define las secciones que se crean en las caras del volumen - Captura de pantalla (Elaboración propia, 2016). . . . .	98
<b>Imagen 2.24.</b>	Algoritmo que define las secciones que se crean en las caras del volumen - Captura de pantalla (Elaboración propia, 2016)	99
<b>Imagen 2.25.</b>	Algoritmo para unir las listas de puntos definidas en las secciones - Captura de pantalla (Elaboración propia, 2016). .	101
<b>Imagen 2.26.</b>	Algoritmo que determina la incorporación de los puntos del lateral AB al listado final - Captura de pantalla (Elaboración propia, 2016) . . . . .	104
<b>Imagen 2.27.</b>	Algoritmo que determina la incorporación de los puntos del lateral CD al listado final - Captura de pantalla (Elaboración propia, 2016) . . . . .	105
<b>Imagen 2.28.</b>	Algoritmo para crear las secciones finales - Captura de pantalla (Elaboración propia, 2016) . . . . .	108
<b>Imagen 2.29.</b>	Algoritmo que determina la incorporación de las laterales AD y BC a las secciones para generar la superficie - Captura de pantalla (Elaboración propia, 2016) . . . . .	111
<b>Imagen 2.30.</b>	Algoritmo utilizado para comprobar la dirección de los vectores definidos por puntos en las secciones y ecuación de la onda - Captura de pantalla (Elaboración propia, 2016) . . .	113
<b>Imagen 2.31.</b>	Algoritmo que desplaza los puntos según la curva de las ondas y define la superficie - Captura de pantalla (Elaboración propia, 2016) . . . . .	117

<b>Imagen 2.32.</b>	Algoritmo que define la triangulación de Delaunay sobre la superficie y exporta la información a Revit - Captura de pantalla (Elaboración propia, 2016) . . . . .	119
<b>Imagen 2.33.</b>	Cuadro Family Types del Módulo A, entorno de trabajo de Revit - Captura de pantalla (Elaboración propia, 2016) . . . .	122
<b>Imagen 2.34.</b>	Cuadro Family Types del Panel A, entorno de trabajo de Revit - Captura de pantalla (Elaboración propia, 2016) . . . . .	122
<b>Imagen 2.35.</b>	Diseño del Módulo A, entorno de trabajo de Revit - Captura de pantalla (Elaboración propia, 2016) . . . . .	123
<b>Imagen 2.36.</b>	Diseño del Panel A, entorno de trabajo de Revit - Captura de pantalla (Elaboración propia, 2016) . . . . .	123
<b>Imagen 2.37.</b>	Representación gráfica del algoritmo generativo, grupos codificados por colores - Captura de pantalla (Elaboración propia, 2016) . . . . .	125
<b>Imagen 2.38.</b>	Resultados del ingreso de diferentes datos en los parámetros modificadores de la base del volumen - Captura de pantalla (Elaboración propia, 2016). . . . .	126
<b>Imagen 2.39.</b>	Resultados del ingreso de diferentes datos en los parámetros modificadores de la cubierta del volumen - Captura de pantalla (Elaboración propia, 2016) . . . . .	127
<b>Imagen 2.40.</b>	Resultados del ingreso de diferentes datos en los parámetros modificadores de las secciones de la superficie - Captura de pantalla (Elaboración propia, 2016) . . . . .	128
<b>Imagen 2.41.</b>	Resultados del ingreso de diferentes datos en los parámetros modificadores de la generación de la superficie - Captura de pantalla (Elaboración propia, 2016) . . . . .	129

<b>Imagen 2.42.</b>	Resultados del ingreso de diferentes datos en los parámetros modificadores de la generación de la superficie y de la ecuación de la onda - Captura de pantalla (Elaboración propia, 2016) . . . . .	129
<b>Imagen 2.43.</b>	Resultados del ingreso de diferentes datos en los parámetros modificadores de la ecuación de la onda - Captura de pantalla (Elaboración propia, 2016). . . . .	130
<b>Imagen 2.44.</b>	Resultados del ingreso de diferentes datos en los parámetros modificadores para dividir la superficie - Captura de pantalla (Elaboración propia, 2016). . . . .	130
<b>Imagen 2.45.</b>	Resultados del ingreso de diferentes datos en los parámetros modificadores para dividir la superficie importados a Revit - Captura de pantalla (Elaboración propia, 2016). . . . .	131

### **CAPÍTULO III 133**

<b>Imagen 3.1.</b>	Máquina de torno CNC - Fotografía (CMZ, s.f.) . . . . .	134
<b>Imagen 3.2.</b>	Máquina de taladro CNC - Fotografía (JPG tools & tools, 2016) . . . . .	134
<b>Imagen 3.3.</b>	Máquina de fresado CNC - Fotografía (DirectINDUSTRY, 2016) . . . . .	135
<b>Imagen 3.4.</b>	Máquina centro de mecanizado CNC - Fotografía (DirectINDUSTRY, 2016) . . . . .	135
<b>Imagen 3.5.</b>	Máquina de cortado láser CNC - Fotografía (DirectINDUSTRY, 2016) . . . . .	135

<b>Imagen 3.6.</b>	Máquina de cortado por chorro de agua CNC - Fotografía (DirectINDUSTRY, 2016) . . . . .	135
<b>Imagen 3.7.</b>	Ventana de selección de procesos para ejecutar del plug-in Dynamo Player para Revit - Captura de pantalla (Elaboración propia, 2016) . . . . .	136
<b>Imagen 3.8.</b>	Resultado de la ejecución del proceso 0. Aplicación del algoritmo generativo dentro de Revit - Captura de pantalla (Elaboración propia, 2016). . . . .	136
<b>Imagen 3.9.</b>	Representación visual del algoritmo 1. Filtración de triángulos problemáticos en Dynamo - Captura de pantalla (Elaboración propia, 2016) . . . . .	137
<b>Imagen 3.10.</b>	Resultado de la ejecución del proceso 1. Filtración de triángulos problemáticos dentro de Revit - Captura de pantalla (Elaboración propia, 2016). . . . .	138
<b>Imagen 3.11.</b>	Representación visual del algoritmo 2. Definición de elementos auxiliares A, B, C en Dynamo - Captura de pantalla (Elaboración propia, 2016). . . . .	139
<b>Imagen 3.12.</b>	Resultado de la ejecución del proceso 2. Definición de elementos auxiliares A, B, C dentro de Revit - Captura de pantalla (Elaboración propia, 2016) . . . . .	140
<b>Imagen 3.13.</b>	Representación visual del algoritmo 3. Codificación de elementos Módulo, Panel y Auxiliares en Dynamo - Captura de pantalla (Elaboración propia, 2016) . . . . .	141
<b>Imagen 3.14.</b>	Resultado de la ejecución del proceso 4. Definición de piezas de sujeción y 5. Codificación de piezas de sujeción dentro de Revit - Captura de pantalla (Elaboración propia, 2016) . . . .	142

<b>Imagen 3.15.</b>	Representación visual del algoritmo 4. Definición de piezas de sujeción en Dynamo - Captura de pantalla (Elaboración propia, 2016) . . . . .	142
<b>Imagen 3.16.</b>	Representación visual del algoritmo 5. Codificación de piezas de sujeción en Dynamo - Captura de pantalla (Elaboración propia, 2016) . . . . .	142
<b>Imagen 3.17.</b>	Representación visual del algoritmo 6. Generación de aberturas en paneles en Dynamo - Captura de pantalla (Elaboración propia, 2016). . . . .	143
<b>Imagen 3.18.</b>	Representación visual del algoritmo 7. Definición de paneles finales en Dynamo - Captura de pantalla (Elaboración propia, 2016) . . . . .	144
<b>Imagen 3.19.</b>	Representación visual del algoritmo 8. Codificación de paneles en Dynamo - Captura de pantalla (Elaboración propia, 2016) . . . . .	144
<b>Imagen 3.20.</b>	Resultado de la ejecución del proceso 6. Generación de aberturas en paneles dentro de Revit - Captura de pantalla (Elaboración propia, 2016). . . . .	145
<b>Imagen 3.21.</b>	Resultado de la ejecución del proceso 7. Definición de paneles finales y 8. Codificación de paneles dentro de Revit - Captura de pantalla (Elaboración propia, 2016) . . . . .	146
<b>Imagen 3.22.</b>	Grabado láser sobre ABS o Acrilonitrilo Butadieno Estireno - Fotografía (CutlaserCut, 2015) . . . . .	149
<b>Imagen 3.23.</b>	Grabado láser sobre PMMA o acrílico - Fotografía (CutlaserCut, 2015) . . . . .	149



<b>Imagen 3.24.</b>	Grabado láser sobre PP o Polipropileno - Fotografía (Cutlaser cut, 2015) . . . . .	149
<b>Imagen 3.25.</b>	Grabado láser sobre Poliestireno - Fotografía (Cutlaser cut, 2015) . . . . .	150
<b>Imagen 3.26.</b>	Grabado láser sobre acero inoxidable - Fotografía (Cutlaser cut, 2015) . . . . .	150
<b>Imagen 3.27.</b>	Grabado láser sobre aluminio - Fotografía (Cutlaser cut, 2015)	150
<b>Imagen 3.28.</b>	Grabado láser sobre espuma de Polietileno - Fotografía (Cutlaser cut, 2015) . . . . .	151
<b>Imagen 3.29.</b>	Grabado láser sobre tejido de lino - Fotografía (Cutlaser cut, 2015) . . . . .	151
<b>Imagen 3.30.</b>	Grabado láser sobre cartón blanco - Fotografía (Cutlaser cut, 2015) . . . . .	151
<b>Imagen 3.31.</b>	Grabado láser sobre caucho de silicona - Fotografía (Cutlaser cut, 2015) . . . . .	152
<b>Imagen 3.32.</b>	Grabado láser sobre plywood - Fotografía (Cutlaser cut, 2015)	152
<b>Imagen 3.33.</b>	Grabado láser sobre MDF - Fotografía (Cutlaser cut, 2015) . .	152
<b>Imagen 3.34.</b>	Máquina de corte vertical para madera - Fotografía (Elaboración propia, 2016). . . . .	155
<b>Imagen 3.35.</b>	Piezas que conforman el módulo 5 - Fotografía (Elaboración propia, 2016) . . . . .	156
<b>Imagen 3.36.</b>	Módulos 7, 8, 9 y 11, previos a su ensamblaje - Fotografía (Elaboración propia, 2016). . . . .	156

<b>Imagen 3.37.</b>	Lámina adhesiva para el módulo 5 - Fotografía (Elaboración propia, 2016) . . . . .	156
<b>Imagen 3.38.</b>	Lámina de adhesivo plástico - Fotografía (Elaboración propia, 2016) . . . . .	156
<b>Imagen 3.39.</b>	Módulo 5 previo a colocar el adhesivo plástico - Fotografía (Elaboración propia, 2016). . . . .	156
<b>Imagen 3.40.</b>	Módulo tras haber colocado el adhesivo plástico - Fotografía (Elaboración propia, 2016). . . . .	156
<b>Imagen 3.41.</b>	Módulo 5 tras haber colocado la lámina de adhesivo plástico - Fotografía (Elaboración propia, 2016) . . . . .	157
<b>Imagen 3.42.</b>	Módulo 3, prototipado, experimentación con uniones con bisagras - Fotografía (Elaboración propia, 2016) . . . . .	157
<b>Imagen 3.43.</b>	Módulo 3, experimentación con uniones con bisagras - Fotografía (Elaboración propia, 2016) . . . . .	157
<b>Imagen 3.44.</b>	Prototipado, experimentación con uniones con bisagras - Fotografía (Elaboración propia, 2016) . . . . .	157
<b>Imagen 3.45.</b>	Módulo 1, experimentación con pega de construcción - Fotografía (Elaboración propia, 2016) . . . . .	158
<b>Imagen 3.46.</b>	Módulos 0 y 1, etapa de prototipado - Fotografía (Elaboración propia, 2016) . . . . .	158
<b>Imagen 3.47.</b>	Módulo 4 con los elementos de sujeción acrílicos - Fotografía (Elaboración propia, 2016). . . . .	158
<b>Imagen 3.48.</b>	Módulo 4 listo para ensamblarse con otros módulos - Fotografía (Elaboración propia, 2016) . . . . .	158

<b>Imagen 3.49.</b>	Detalle del elemento de sujeción entre los paneles 0c y 1b de los módulos 0 y 1 - Fotografía (Elaboración propia, 2016) . .	158
<b>Imagen 3.50.</b>	Elementos de sujeción entre módulos - Fotografía (Elaboración propia, 2016) . . . . .	158
<b>Imagen 3.51.</b>	Módulos 0 y 1 ensamblados entre sí - Fotografía (Elaboración propia, 2016) . . . . .	159
<b>Imagen 3.52.</b>	Módulos 2, 3, 7, 8 y 9 ensamblados, parte frontal - Fotografía (Elaboración propia, 2016). . . . .	159
<b>Imagen 3.53.</b>	Módulos 2, 3, 7, 8 y 9 ensamblados, parte posterior - Fotografía (Elaboración propia, 2016). . . . .	159



Sebastián Andrés Rodas Rivera, del trabajo de titulación de tipo exploratorio y diseño experimental "INTRODUCCIÓN AL DISEÑO PARAMÉTRICO: UTILIZACIÓN DE HERRAMIENTAS DIGITALES PARA LA RESOLUCIÓN DE SUPERFICIES COMPLEJAS MEDIANTE LA APLICACIÓN DE ALGORITMOS", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Arquitecto. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, marzo de 2017

SEBASTIAN RODAS

Sebastián Andrés Rodas Rivera

C.I: 0103730024



Sebastián Andrés Rodas Rivera, autor del trabajo de titulación de tipo exploratorio y diseño experimental "INTRODUCCIÓN AL DISEÑO PARAMÉTRICO: UTILIZACIÓN DE HERRAMIENTAS DIGITALES PARA LA RESOLUCIÓN DE SUPERFICIES COMPLEJAS MEDIANTE LA APLICACIÓN DE ALGORITMOS", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, marzo de 2017

SEBASTIAN RODAS

Sebastián Andrés Rodas Rivera

C.I: 0103730024



Cristian Jhovanny Benavides Padilla, del trabajo de titulación de tipo exploratorio y diseño experimental "INTRODUCCIÓN AL DISEÑO PARAMÉTRICO: UTILIZACIÓN DE HERRAMIENTAS DIGITALES PARA LA RESOLUCIÓN DE SUPERFICIES COMPLEJAS MEDIANTE LA APLICACIÓN DE ALGORITMOS", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Arquitecto. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, marzo de 2017

---


Cristian Jhovanny Benavides Padilla

C.I.: 0302066246



Cristian Jhovanny Benavides Padilla, autor del trabajo de titulación de tipo exploratorio y diseño experimental "INTRODUCCIÓN AL DISEÑO PARAMÉTRICO: UTILIZACIÓN DE HERRAMIENTAS DIGITALES PARA LA RESOLUCIÓN DE SUPERFICIES COMPLEJAS MEDIANTE LA APLICACIÓN DE ALGORITMOS", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, marzo de 2017

  
Cristian Jhovanny Benavides Padilla

C.I: 0302066246

## **DEDICATORIA**

Quiero dedicar este trabajo a toda mi familia, mis padres, Luis Rodas y Catalina Rivera, que han dado todo de sí mismos para que nunca me falte nada, por el amor que me han dado, la educación y todos los valores que me han inculcaron y me han hecho la persona que soy hoy, mis hermanos, Juan Luis Rodas, Pedro Rodas y Ma. Caridad Rodas, quienes han sido un ejemplo y una fuente de inspiración para mí, a mis tías Raquel Rivera y Elena Rivera a quienes amo mucho y son gran parte de mi vida.

De manera muy especial le dedico este trabajo y el logro que cumpla con este, a mi esposa Paola Castillo y mi hija Simoné Rodas, quienes son la razón por la cual se justifica todo el esfuerzo que he puesto en este trabajo, Uds. son mi inspiración, mis amores.

Sebastián Andrés Rodas Rivera



## **DEDICATORIA**

La presente tesis está dedicada a toda mi familia y amigos, principalmente a mis padres, Rubén Benavides y Magdalena Padilla por el apoyo que siempre me brindaron en todo el transcurso de mi carrera universitaria, todo este trabajo ha sido posible gracias a ellos.

Al Arq. Raúl Cordero, gracias por su tiempo compartido y por impulsar el desarrollo de este trabajo de titulación.

Cristian Jhovanny Benavides Padilla

## AGRADECIMIENTOS

Queremos agradecer primero y sobre todo a Dios por permitirnos llevar a cabo este trabajo, sin Su bendición no hubiese sido posible la culminación del presente trabajo.

Agradecemos a nuestras familias, quienes nos han brindado su apoyo y comprensión durante todo el trayecto de la vida universitaria y de especial manera en esta etapa final.

De manera especial al Arq. Boris Adrián Orellana Alvear por habernos brindado su amistad y por el interés puesto para que el presente trabajo llegue a su culminación, por toda la ayuda prestada por su parte desde el inicio de este trabajo.

Al Arq. Raúl Esteban Cordero Gulá, director del trabajo de titulación, quién ha sabido guiarnos y expresar su visión crítica y constructiva sobre la realización del trabajo, por sus consejos y conocimientos mediante los cuales ha sido posible la finalización del presente trabajo.

Finalmente queremos expresar nuestro agradecimiento de manera especial a la Universidad de Cuenca y a todos los docentes que han participado en nuestra formación profesional, por todos los conocimientos brindados con la única expectativa de formarnos como profesionales de calidad moral y ética.

## PREFACIO

Dentro de la arquitectura moderna el diseño paramétrico resulta como una especie de paradigma en lo relacionado al diseño, en el cual la relación entre los elementos se utiliza para manipular y definir el diseño, resultando así en geometrías y estructuras complejas.

Al diseño paramétrico se le puede considerar como una herramienta para el desarrollo creativo (Woodbury, 2010) o como un estilo arquitectónico completamente nuevo (Schumacher, Parametricism Manifesto, 2008). Este tipo de diseño que posee un enfoque paramétrico es evaluado como un recurso innovador que puede ser aplicado a diferentes escalas, por ejemplo, en diseño industrial, estructural, arquitectónico e inclusive en urbanismo.

En el presente estudio se partirá de un espacio arquitectónico existente, dentro del cual se generará un espacio de expresión diferente al original mediante la utilización de herramientas digitales y más específicamente el diseño paramétrico.

La aparición de programas de diseño CAD (Diseño Asistido por Computadora o Computer Aided Design por sus siglas en inglés) que cada vez son capaces de realizar cálculos más precisos y sofisticados, junto con la evolución tecnológica de los materiales para la construcción, han contribuido a la aparición de nuevas formas de construir. Es así que este proyecto utilizará una de las herramientas más potentes dentro de lo que se refiere al CAD, el software de programación visual.

El estudio proyectará un diseño paramétrico que estará basado en los algoritmos generativos como elementos generadores de espacios y arquitectura, el algoritmo resultante será la representación de una serie de instrucciones que se le dará al computador a manera de un diagrama de flujos, mediante el uso de programación visual no será necesario tener conocimientos sobre lenguajes de programación. Además, se establecerá una base en la cual el diseño paramétrico es una herramienta útil para la labor del diseño arquitectónico, que posee diferentes recursos computacionales, ya sea para generar o computar geometrías, desarrollar análisis técnicos o simplemente ayudar en la documentación de los diferentes proyectos.





## CAPÍTULO I

### 1. ANTECEDENTES Y MARCO TEÓRICO

#### 1.1. ASPECTOS GENERALES

##### 1.1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad de cierta manera se ha admitido ampliamente que la arquitectura contemporánea, al igual que otras áreas de las actividades humanas, se encuentran dominados por las computadoras y el paradigma de la computación, al principio las computadoras sirvieron como herramientas para facilitar el trabajo manual, actualmente las computadoras se utilizan para calcular y racionalizar el procesamiento de la información (Khabazi, 2012).

La aparición del CAD (Diseño Asistido por Computadora) fue uno de los avances tecnológicos más importantes en la práctica del diseño, mediante el uso del software CAD fue posible afrontar problemas de orden geométrico más complejos, posteriormente el software CAD adquiriría acceso al scripting y herramientas algorítmicas.

Con esto se llega a tener la capacidad de “codificar” la geometría generada consiguiendo así un gran potencial de diseño. El Diseño Asistido por Computadora evolucionó en Diseño Generativo (Benedetti, 2012).

La aplicación del diseño paramétrico dentro del campo de la arquitectura se ha centrado o enfocado casi en su totalidad a obras de gran escala y dentro de nuestro medio. La práctica común del diseño arquitectónico no utiliza toda la capacidad actual de las herramientas digitales para facilitar los procesos de diseño y construcción de formas complejas, las mismas que pueden ser incorporadas al proyecto o que inclusive pueden llegar a ser instrumentos generadores.

La falta de experiencias en nuestro medio local en lo referente al diseño paramétrico contrasta de gran manera con la amplia cantidad de práctica y experimentación que se ha desarrollado en el ámbito internacional, se hace evidente la necesidad de realizar un estudio que nos acerque al contexto del diseño paramétrico, sus capacidades y limitaciones.

#### 1.1.2. OBJETIVO DE LA INVESTIGACIÓN

##### 1.1.2.A. OBJETIVO GENERAL

Crear un documento teórico-práctico, que sirva como base y material de investigación sobre el diseño paramétrico, especialmente enfocado en la resolución de superficies complejas y la posterior fabricación de piezas, mediante el uso de algoritmos y software especializado.

##### 1.1.2.B. OBJETIVOS ESPECÍFICOS

- Estudiar las herramientas digitales que se utilizan en la actualidad para el diseño paramétrico, realizar una comparación para conocer sus capacidades y escoger la plataforma más adecuada para la realización del trabajo, según los criterios de los autores.
- Establecer una idea general para el diseño de la superficie compleja y establecer las relaciones matemáticas y geométricas para la creación del algoritmo generativo
- Crear los algoritmos necesarios para la generación de la superficie compleja y el proceso de fabricación, mediante los conceptos aprendidos en la investigación.
- Realizar una maqueta a escala 1:1, de una porción de la superficie total, la misma que servirá como muestra de un todo.

##### 1.1.3. JUSTIFICACIÓN

En la actualidad la práctica del diseño arquitectónico podría verse potenciada gracias a la utilización de los diferentes avances tecnológicos que se han desarrollado, se puede así advertir que los procesos de cambio son cada vez más acelerados, las herramientas digitales de

última generación permiten que esto sea posible en aspectos operativos.

Los tiempos totales que demandaba cualquier proyecto, en cuanto a la elaboración de documentación y diseño, han sido optimizados en gran parte por el uso de las tecnologías digitales. Las herramientas digitales enfocadas hacia el diseño paramétrico facilitan la creación de formas y geometrías complejas tanto en su etapa de diseño, como en su construcción. Es así que se propone la aplicación del diseño paramétrico para generar un espacio mediante el uso de algoritmos y software CAD.

#### 1.1.4. PREGUNTAS DE INVESTIGACIÓN

Dado el nivel actual de utilización de las computadoras como herramientas en casi todas las actuaciones humanas, y de manera especial para nuestro estudio en el campo del diseño y la arquitectura, el presente trabajo busca dar respuesta a las siguientes preguntas de investigación:

- ¿Qué capacidad para su uso y aplicación tienen estas herramientas?
- ¿Qué fortalezas y debilidades tiene el diseño paramétrico durante todo el proceso?
- ¿Hasta qué punto se puede ocupar esta práctica de diseño en nuestro medio?

- ¿Qué limitaciones se pueden presentar tanto por los materiales utilizados como por la maquinaria requerida para su puesta en obra y manufactura?

Siendo así, los diferentes resultados que se conseguirán mediante el presente estudio podría derivarse en la aplicación de diferentes estrategias que se acoplen a nuestro medio en el uso de materiales y maquinaria especializada para la manufactura de las piezas y posterior puesta en obra.

#### 1.1.5. HIPÓTESIS

Las herramientas digitales seleccionadas son capaces de crear algoritmos que permiten la automatización de manera iterada de procesos de diseño y fabricación

que pueden resultar muy complejos y tediosos al realizarse manualmente.

### 1.2. ESTADO DEL ARTE

#### 1.2.1. ANTECEDENTES

Desde las pirámides antiguas a las construcciones actuales, los edificios han sido diseñados y construidos en relación a una variedad de factores cambiantes, incluyendo el clima, la tecnología, el uso, entre otros. El diseño paramétrico no pretende redefinir la arquitectura o la profesión, sin embargo, ofrece una valiosa herramienta que ha permitido a arquitectos diseñar y construir edificios innovadores, con condiciones cualitativas y cuantitativas más exigentes (AIACC, 2012).

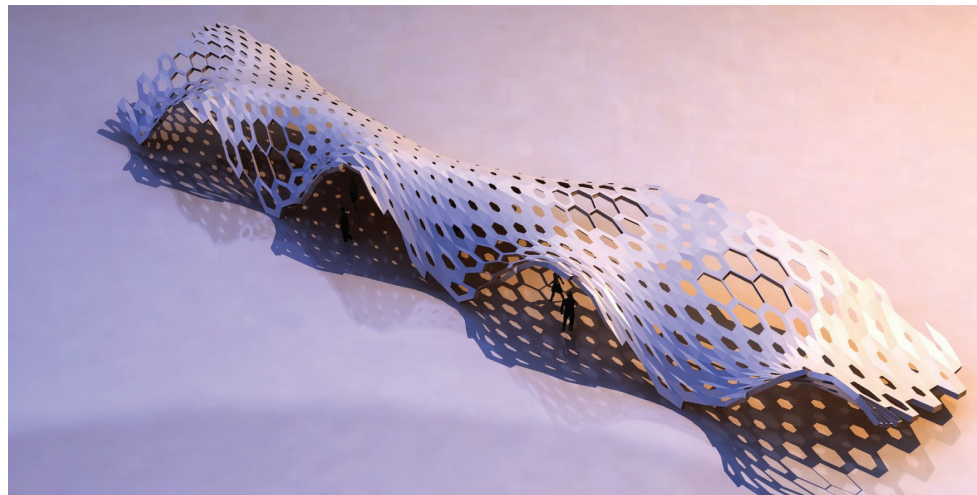


Imagen 1.1. Estructura paramétrica - Imagen digital (Espacio Arquitectura, 2013)

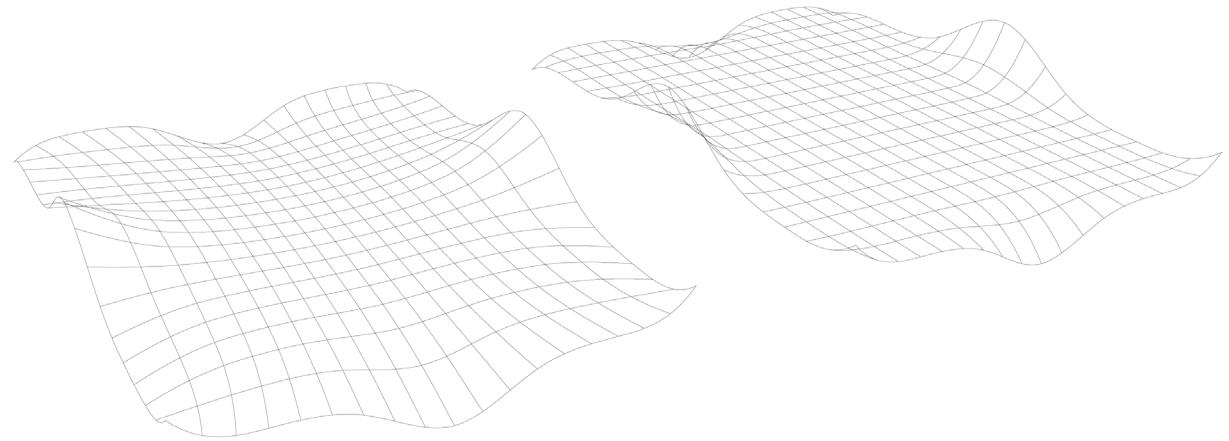


En los siglos XIX y XX los grandes cambios que se han dado en los campos de la ingeniería y la arquitectura fueron posibles gracias al uso de nuevos materiales y sistemas estructurales, mientras que en el siglo XXI estos cambios están dirigidos por la capacidad de los sistemas digitales para desarrollar y manejar geometrías complejas, lo que a su vez establece nuevos paradigmas en el ámbito de la arquitectura.

En 1964, en una conferencia llevada a cabo por el Centro de Arquitectura de Boston, se hizo evidente que la era electrónica tendría un efecto dramático en el diseño de edificios; las industrias aeroespaciales estaban usando el poder computacional de los ordenadores para calcular superficies deformadas y complejas simulaciones de trayectorias de vuelo, esto fascinó a los arquitectos.

En la década de 1980 los arquitectos y los diseñadores comenzaron a usar las computadoras que ejecutan el software desarrollado para la industria aeroespacial con la finalidad de “animar la forma”.

En 1999 el arquitecto de Los Ángeles Greg Lynn se convirtió en el principal teórico y diseñador en poner en práctica la teoría, al usar el ordenador para generar lo que se convirtió en su notoria arquitectura “blob” y “folding”. En su libro “Animate Form” estudió la historia y establece las directrices para la arquitectura que se podría generar



**Imagen 1.2.** Superficies deformadas - Imagen digital (Parametric Wood, 2011)

mediante el cálculo de sistemas genéticos y códigos, aunque sólo sea virtualmente en el ordenador. La “Spline” demostró ser la forma más relevante por su sencilla y concisa capacidad paramétrica.

En la Universidad de Princeton en 2002, se incorporan los principios de diseño paramétrico mediante la creación de una superficie, que tiene relación con el movimiento de una corriente oceánica (AIACC, 2012).

El avance actual en el tema de software dedicado al diseño arquitectónico ha permitido de cierta manera un cambio en el paradigma actual del proceso de la arquitectura, esta evolución de las herramientas digitales ha llevado a la posibilidad de modelar y calcular formas

y estructuras complejas dentro de la práctica arquitectónica lo cual no era posible anteriormente (Benedetti, 2012).

En la actualidad los procesos de cambio son cada vez más acelerados, y las actuales herramientas tecnológicas permiten que esta aceleración sea posible en aspectos operativos. Los tiempos reales que demandaban determinados proyectos han sido disminuidos dependiendo de los requerimientos de cada proyecto gracias a la aplicación de herramientas digitales.

De igual manera que el diseño empírico, que básicamente se fundamenta en la experimentación y análisis de los resultados, en una especie de proceso de prueba y error, lo paramétrico apunta precisamente a la posibilidad de experimentación, del

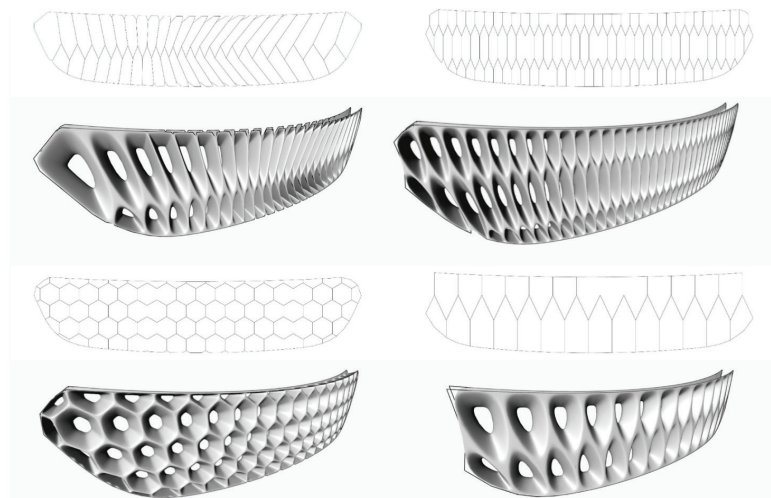


juego entre los valores de los parámetros, que a su vez son capaces de producir un número casi infinito de posibilidades, pero precisamente como lo empírico ha pasado por un proceso hasta llegar a un objetivo deseado, el tema de lo paramétrico se encuentra recién en sus primeros momentos, al menos dentro de nuestro medio.

Aparece entonces el patrón como un elemento que compone un todo, es aquí que el diseño paramétrico puede realmente apreciar su potencial, en la capacidad de generar un todo en base a un solo elemento definido, este patrón que puede ser aplicado tanto a escalas pequeñas, como a escalas gigantes, una ciudad, por ejemplo.

Los patrones son cosas tan flexibles y maleables en su configuración que, para un mismo problema o diseño, se pueden presentar un número tan grande de variables que no se llegará a explorar todas las posibles opciones (Navarrete, 2014).

Dentro del diseño paramétrico han destacado tres movimientos principalmente, estos son el Posmodernismo, el Deconstructivismo y el Folding, el estilo que engloba todo aquello referente al diseño paramétrico se lo conoce como parametricismo, se habla del Posmodernismo y la Deconstrucción como movimientos de transición (Benedetti, 2012).



**Imagen 1.3.** T-splines en Dynamo - Imagen digital (Kron, 2016)

### 1.3. MARCO TEÓRICO

La tecnología informática ha dado a los diseñadores y arquitectos las herramientas para analizar y simular la complejidad observada en la naturaleza y aplicarla a formas estructurales de construcción y los mecanismos de organización urbana.

#### 1.3.1. DISEÑO PARAMÉTRICO

##### 1.3.1.A. PRECURSORES FILOSÓFICOS DEL PARAMETRICISMO

En los diferentes textos consultados para la realización de esta investigación, resaltan sobre todo tres nombres principales en lo referente a personajes que han influenciado en gran manera a la conceptualización actual del parametricismo y han realizado grandes aportes filosóficos para el posterior pensamiento paramétrico, estos son: Antoni Gaudí, Frei Otto y Greg Lynn.

##### Antoni Gaudí y la Sagrada Familia

Es conocido que a Gaudí no le gustaba dibujar, es por esto que utilizaba un proceso de exploración de formas muy diferente al tradicional.

La iglesia de la Sagrada Familia es considerada como un paradigma en el campo de la fabricación digital, la manera en la cual desarrolló el proyecto es a través de la creación de modelos físicos realizados en obra.





Gaudí utilizó curvas catenarias paramétricas y paraboloides hiperbólicas paramétricas, esto es mejor ilustrado en la representación de su modelo de cadenas colgantes.

Al estar sujetas las cadenas al techo su propio peso, y a veces un peso extra adherido, crearía una curva catenaria, que, según la ley de Hooke, se encontraría siempre en tracción pura, al invertir el modelo, y mantener la curva, el resultado sería un elemento que se encuentra siempre en compresión pura, al cambiar ciertos parámetros en el modelo, los resultados variarían de forma acorde (Makert & Alves, 2016).

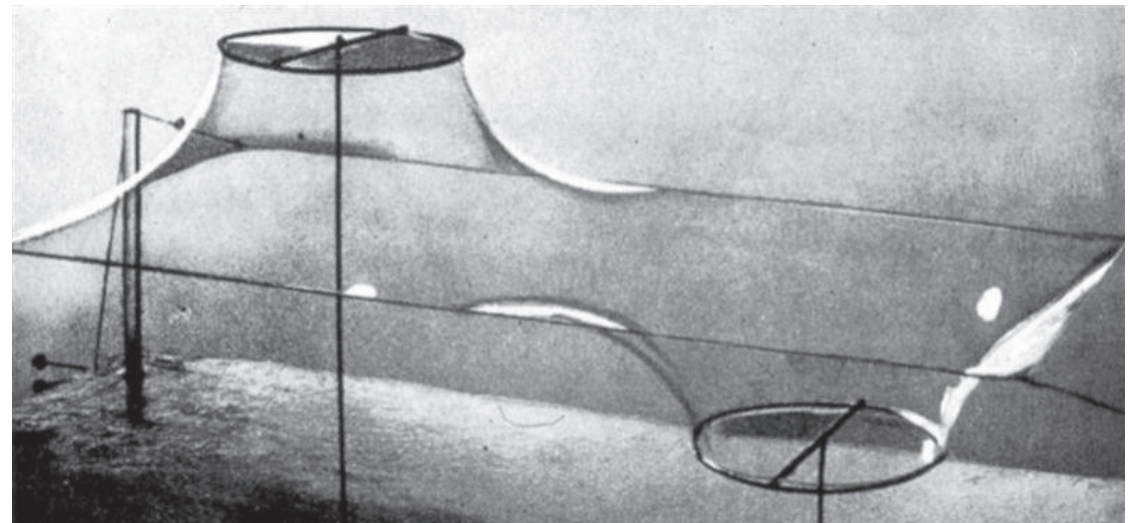
Mediante la experimentación con modelos físicos de formas complejas, el conocimiento que resultante de este proceso empírico, le permitió a Gaudí crear una de las obras de arquitectura más sorprendentes e intrigantes de todos los tiempos, sin contar con las herramientas digitales de la actualidad, esto recalca a la parametría como el proceso de encontrar una forma, más que la forma en sí.

### **Frei Otto y el proceso de búsqueda de formas**

Gracias a la aplicación de técnicas de diseño paramétricas y algorítmicas, la síntesis de lo vernacular se vuelve tendencia, sobre todo en lo referente a los patrones urbanos y naturales.



**Imagen 1.4.** Modelo de cuerdas colgadas con pesos de Gaudí - Fotografía (Architizer, 2016)



**Imagen 1.5.** Experimentación de superficies mediante el uso de pompas de jabon, Frei Otto - Fotografía (Plataforma Arquitectura, 2015)

Otto explica los fenómenos de las redes urbanas como sistemas que se organizan a sí mismos, todo el crecimiento estructural de la ciudad se basa en dos principios básicos que determinan y organizan todos los espacios humanizados: la ocupación y la conexión.

Estos principios se basan en leyes de atracción y repulsión, también se puede entender como expansión o contracción, estos modificadores urbanos presentan un comportamiento similar al que se encuentra en los patrones naturales y orgánicos.

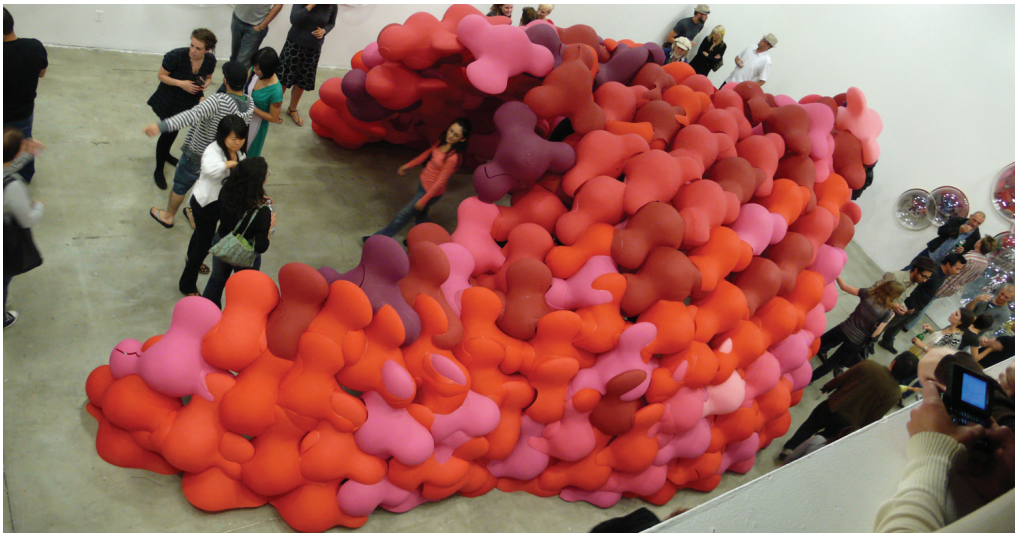
Los fenómenos explicados anteriormente se han ilustrado mediante una serie de experimentos de búsqueda de forma.

- Repulsión u ocupación distante: Se realizaron experimentos con agujas magnetizadas, el balance entre las fuerzas magnéticas crea rejillas triangulares, las mismas que incrementan su regularidad conforma incrementa el número de agujas.
- Ocupación atrayente: Se utilizaron aparatos que crean burbujas de jabón o aire de dimensión constante, las mismas que se organizan entre sí óptimamente de manera hexagonal, el proceso de formación del patrón es similar al expuesto anteriormente.
- Ocupación atrayente y repulsiva simultánea: Este modelo combina los dos experimentos anteriores, se utilizan

pequeñas burbujas o partículas de poliestireno en conjunto con agujas magnetizadas, el resultado demuestra los dos fenómenos, atracción y repulsión.

- Sistema de trayectoria directa: En este sistema cada par de puntos se unen mediante la geodésica, la misma que se define como la menor longitud existente entre dos puntos y que pertenece a una superficie dada, no existen ramificaciones.
- Sistema de trayectoria mínima: Este sistema pretende conectar un conjunto de puntos a través de la ruta más corta posible, para esto se utiliza el concepto del árbol Euclidiano de Steiner.
- Sistema de trayectoria con desvío minimizado: Este sistema combina las ventajas de los dos sistemas anteriores, al mismo tiempo que se optimiza la longitud y la desviación, cada proceso de experimentación produce una forma distinta (Lopes, Paio, & Sousa, 2014).

Todos los modelos descritos anteriormente son sin duda ejemplos de la aplicación de parámetros, determinados para la resolución de superficies y trayectorias complejas, pero que se presentaban de manera habitual en la naturaleza.



**Imagen 1.6.** Pavellón Blobwall de Greg Lynn - Fotografía (Revista PLOT, 2012)





Todo esto se desarrollaba de manera intuitiva mediante la exploración de los resultados de los experimentos, mucho antes de que las herramientas digitales estuvieran al alcance o nivel en el que se encuentran hoy.

### Greg Lynn y la Blobitectura

Greg Lynn es un innovador en lo referido a redefinir el medio de diseño hacia la tecnología digital, al igual en utilizar los métodos CNC para fabricar formas complejas funcionales y ergonómicas.

Gracias a sus estudios en arquitectura y filosofía se ha involucrado en combinar las realidades del diseño y la construcción con el potencial especulativo de la escritura y enseñanza (GREG LYNN FORM, s.f.).

El termino BLOB hace referencia a Binary Large Object, u Objeto Binario Grande en español, la arquitectura BLOB o Blobarquitectura como la llama Lynn, hace referencia a un tipo de arquitectura con formas orgánica, que no posee bordes rectos o formas simétricas, se presentan construcciones con ondas y curvas sobre sus superficies, todo esto siendo posible gracias a la utilización del software CAD.

Se entiende a la arquitectura BLOB como una representación anterior al parametricismo, que se enfoca más en las relaciones matemáticas y geométricas que existen entre los objetos (Fuentes, 2016).

### 1.3.1.B. DEFINICIÓN DE PARÁMETRO

Se entiende al parámetro como una variable de diferenciación, es capaz de definir los límites de un sistema y las condiciones necesarias para su funcionamiento. Es aquello que nos permite integrar y crear relaciones lógicas entre las partes de un todo.

Mediante la asunción de que todo puede ser cuantificado, todo puede por ende ser parametrizado, el parámetro implica una formula, en la cual, al modificarse el valor del parámetro, producirá un resultado diferente (Benjamin, y otros, 2014).

Sin embargo, la manera en la cual se utilizan los parámetros es esencial para el resultado de un proceso.

Por ejemplo, en lugar de utilizar los parámetros para generar formas mediante un proceso, se los utiliza para estudiar todas las posibles formas resultantes de dicho proceso, para posteriormente poder maximizar una serie de cualidades de desempeño basadas en una serie de parámetros que se encontraban ocultos, tal vez porque la interacción con el modelo no se encontraba lo bastante clara (Benjamin, y otros, 2014).



Imagen 1.7. Proceso de construcción del Heydar Aliyev Center - Fotografía (Adelto, 2013)

### 1.3.1.c. El Parametricismo como Estilo Arquitectónico

Según Patrick Schumacher (Schumacher, Parametricism, 2009), recientemente se ha establecido casi con una aceptación extendida en la arquitectura de "avant-garde" al parametricismo como un nuevo estilo.

Se explica que el Parametricismo tiene sus raíces en técnicas de animación digital, siendo refinado por los sistemas de diseño paramétrico avanzado y los métodos de scripting.

Se ha desarrollado a lo largo de los últimos 15 años, y pretende ser un estilo sucesor al Modernismo, además de acabar con la incertidumbre de una fase marcada por algunos episodios arquitectónicos cortos, como el Post modernismo, Deconstructivismo y Minimalismo.

Se ha extendido en hacia todas las escalas de la arquitectura, desde el diseño interior, hasta la escala masiva del diseño urbano, es más, Schumacher (Schumacher, Parametricism, 2009) afirma que mientras más grande el proyecto, se pueden evidenciar de manera más clara las ventajas y capacidad superior que tiene el Parametricismo para articular la complejidad del proyecto (Schumacher, Parametricism, 2009).



**Imagen 1.8.** Heydar Aliyev Center, representante del deconstructivismo - Fotografía (Deconstructivismo, 2015)

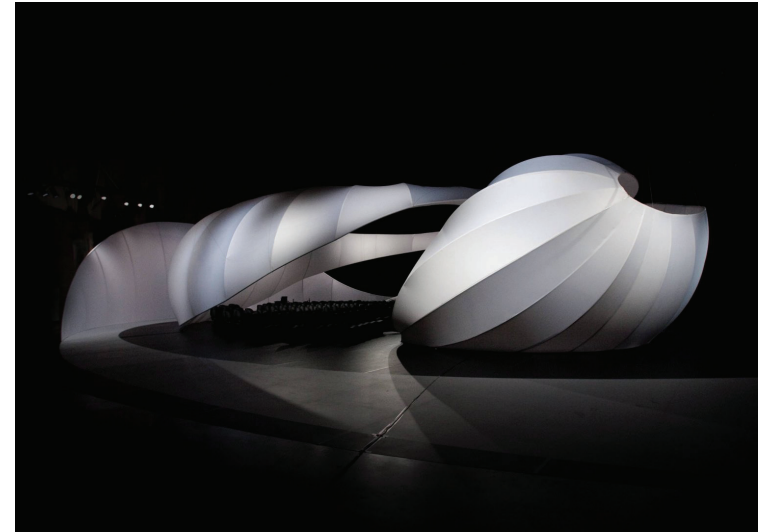


**Imagen 1.9.** Museo de Arte de Tel Aviv, representante del folding - Fotografía (A-cero Blog, 2015)





**Imagen 1.10.** Guangzhou Opera House, exterior - Fotografía (Equipo Arquidesign, 2011)



**Imagen 1.12.** JS Bach Chamber Music Hall, exterior - Fotografía (DailyTonic, 2012)



**Imagen 1.11.** Guangzhou Opera House, interior - Fotografía (LC&LP, 2016)



**Imagen 1.13.** JS Bach Chamber Music Hall, interior - Fotografía (Aminta Online, 2010)

Se busca lidiar con la nueva demanda de arquitectura con el uso de técnicas de diseño paramétrico, las técnicas en sí, el uso de animación, herramientas de simulación y modelado paramétrico, han inspirado un nuevo movimiento colectivo con nuevas ambiciones (Schumacher, Parametricism, 2009).

Si bien muchos de los componentes de un diseño específico pueden reutilizarse directamente para otro diseño, otros varían en cuanto a dimensiones, pero manteniendo su forma. Los programas CAD con capacidad paramétrica permiten alterar la forma de un modelo modificando los valores numéricos de sus dimensiones. De esta manera es posible flexibilizar enormemente el proceso de

diseño y mantener bibliotecas de formas básicas que pueden emplearse de forma versátil en diferentes diseños. (Alvarez & Roces, 2005).

El diseño paramétrico está basado en un esquema algorítmico que permite expresar parámetros y reglas que definen, codifican y aclaran la relación entre los elementos que conforman el diseño y el diseño resultante (Wassim, 2013).

El diseño paramétrico es parte de un nuevo paradigma en la investigación proyectual ya que permite al diseñador crear sus propias herramientas, generando geometrías a partir de parámetros establecidos al inicio del proceso y un conjunto de relaciones formales.



**Imagen 1.14.** Fabricación y ensamblaje del diseño paramétrico - Fotografía (Parametric Fabrication, 2011)

Esta metodología permite no sólo crear un diseño, sino una infinidad de instancias del mismo diseño a partir de la alteración de la programación inicial.

Los objetos paramétricos son aquéllos definidos por ciertas variables que se relacionan por medio de ecuaciones paramétricas. Los vínculos y relaciones que establecen con otros objetos se ordenan según una estructura en árbol que puede ser modificada. De la misma forma, es posible configurar las partes que componen un objeto que se mantienen relacionadas entre sí.

El trabajo con estas variables se convierte ahora en el fundamento del proyecto arquitectónico. No se trata ya del hecho de imaginar una forma determinada, sino de tener la capacidad de concebir y diseñar las relaciones y reglas que articulan entre sí los objetos, familias y partes componentes de un proyecto arquitectónico.

Como los exponentes más reconocidos dentro de este movimiento tenemos a Patrik Schumacher (quien aportó de gran manera en la parte teórica, es director fundador del Design Research Laboratory at the Architectural Association School of Architecture, socio de Zaha Hadid desde 1988), Neil Spiller (arquitecto, artista y escritor, también es director de AVATAR, Advanced Virtual and Technological Architecture Research, el cual es un laboratorio dedicado a investigar el impacto



que la tecnología avanzada tiene sobre el diseño arquitectónico), Rem Koolhaas (de gran aporte proyectual y conceptual), Diller Scofidio, Coop Himmelblau, Peter Eisenman, Lebbeus Woods, Zaha Hadid, Morphosis (Thom Mayne) y Frank Gehry; siendo los últimos tres arquitectos aquellos que han generado de mayor manera un cambio en el ámbito de la arquitectura a nivel mundial (Benedetti, 2012).

Existen dos tipos principales de sistemas de modelado paramétrico:

- Los sistemas de propagación, basados en calcular las incógnitas a partir de determinados datos utilizando un modelo de flujo de datos.
- Los sistemas de restricción, que resuelven conjuntos de restricciones continuas y discretas.

La investigación de la forma, es una de las estrategias de aplicación de un sistema basado en la propagación. La idea detrás de la investigación de la forma es optimizar ciertos objetivos de diseño teniendo en cuenta un conjunto de restricciones o condiciones generales que deben ser satisfechas por el diseño. (Woodbury, 2010).

El diseño paramétrico se enfoca en crear elementos que puedan ser modificados mediante la asignación de un conjunto de relaciones entre los mismos, los resultados

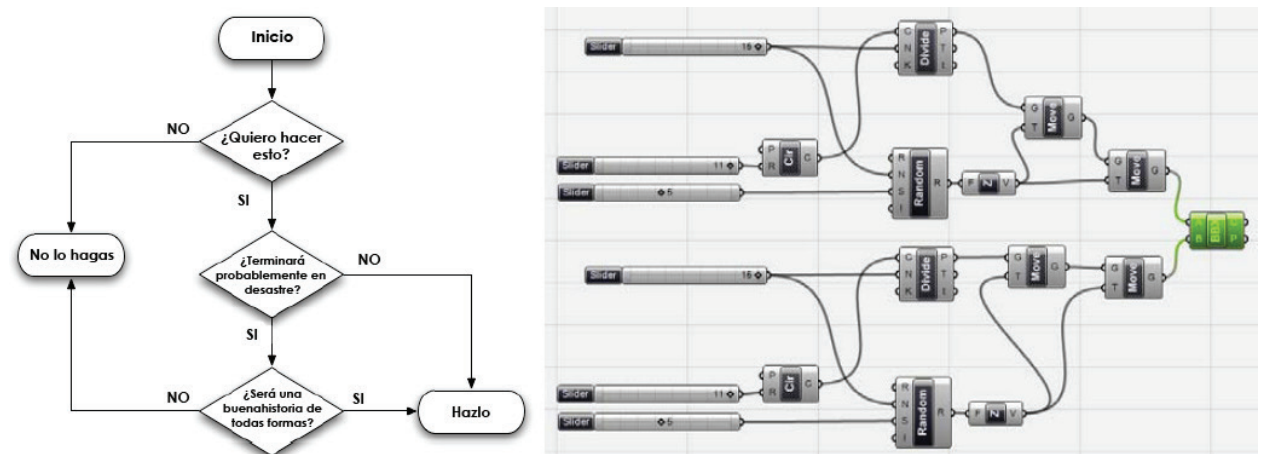
al final del proceso pueden ser analizados para definir si cumplen o no con lo pensado por el diseñador.

La capacidad intrínseca que posee el diseño paramétrico está más que nada enfocada en la posibilidad de experimentar con los resultados, cambiar valores y comparar el nuevo resultado con el anterior, sin la necesidad de repetir el trabajo.

El diseño paramétrico permite la configuración de la geometría y los diferentes parámetros que la rigen, por ejemplo, el ancho de un rectángulo, la altura de un triángulo, o el radio de una circunferencia, de la misma manera,

en los parámetros se puede controlar la ubicación de cada elemento en el espacio xyz, el usuario debe entonces interactuar con los diferentes modificadores para poder establecer el resultado deseado.

Se contemplan diferentes alternativas para llegar al resultado deseado, por lo general los proyectos se realizan de tal manera que siguen un orden específico preestablecido, una serie de instrucciones que se adaptan para llegar al resultado final, mientras que en el diseño paramétrico se pretende crear elementos que se adapten a las condiciones lógicas que se les asigna, esto varía según los requisitos de cada proyecto.



**Imagen 1.15.** Esquema de flujo vs. algoritmo generado en Grasshopper - Imagen digital (GeometricMind, 2011)



### 1.3.1.D. SOFTWARE Y DISEÑO PARAMÉTRICO

Los parámetros son como una especie de propiedades que comprenden un elemento, y se utilizan para modificar la manera en la que se comporta el elemento, puede por ejemplo determinar su forma, ubicación o composición.

La arquitectura paramétrica se puede entender como la aplicación de parámetros que modifican el comportamiento de los elementos que conforman el diseño arquitectónico, ciertas tareas se pueden automatizar gracias a la aplicación de herramientas desarrolladas para este fin, como Grasshopper o Dynamo.

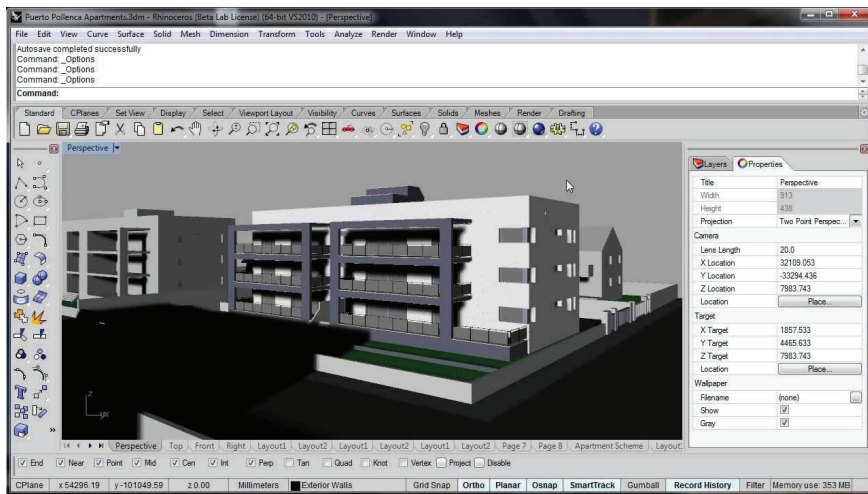
Lo más probable es que la mayoría de los arquitectos estén al tanto de Rhinoceros como software que permite la modelación en 3D de manera intuitiva y precisa, sin embargo, Grasshopper resulta ser un software más nuevo o desconocido. Grasshopper es un plug-in para Rhinoceros, orientado al diseño paramétrico que funciona como editor de algoritmos generativos.

Las ventajas de este programa son que, a diferencia de muchos, con este no se necesita experiencia en programación o scripting, lo cual permite crear diseños paramétricos a partir de componentes generadores y modificadores, obteniendo una considerable optimización de tiempo (Molinare, 2011).

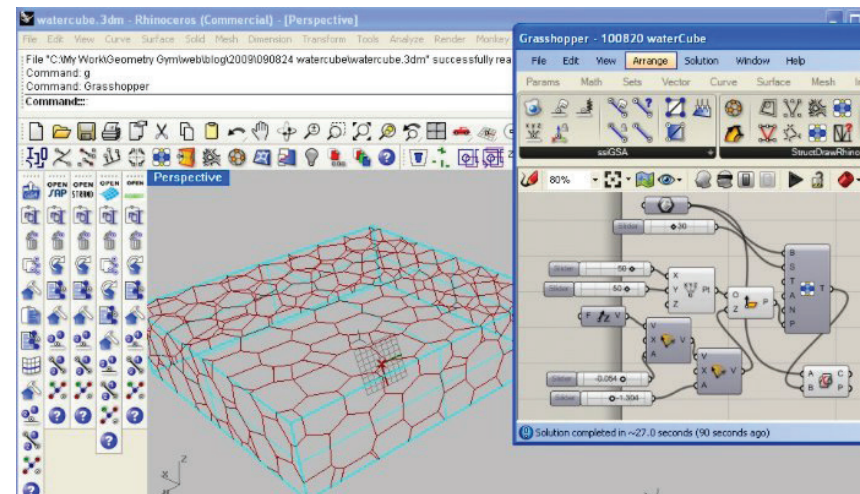
Otros programas o softwares utilizados ampliamente en el medio para el desarrollo del diseño paramétrico son Autodesk Revit, CATIA, Autodesk 3DS Max, Autodesk Maya, Autodesk Dynamo Studio, Marionette, Modelur y GenerativeComponents.

Autodesk Revit es una plataforma de diseño y documentación que soporta los diseños, dibujos y tablas de planificación que son necesarias para el Modelado de Información de Edificios (BIM, por sus siglas en inglés). El BIM es capaz de procesar información sobre el diseño del proyecto, cantidades, y las diferentes etapas cuando se necesite.

El modelado paramétrico hace referencia a las relaciones entre los elementos en un



**Imagen 1.16.** Entorno de trabajo de Rhinoceros - Captura de pantalla (Simply Rhino Rhino3D Channel, 2012)



**Imagen 1.17.** Entorno de trabajo de Grasshopper para Rhino - Captura de pantalla (MBWAREZ, 2011)





proyecto que permiten la coordinación y el manejo de los cambios que Revit provee. Estas relaciones son creadas automáticamente por el software o por uno mismo mientras se trabaja en el proyecto.

Autodesk 3DS Max es un software de modelado 3D paramétrico que permite el modelado, animación y renderizado para juegos, películas y gráficos en movimiento. El software utiliza el concepto de modificadores conectados por cables a los parámetros que controlan la geometría y le permite al usuario la habilidad de codificar su funcionalidad. Es similar al ambiente de trabajo para Grasshopper y Dynamo.

Autodesk Maya es un software de gráficos 3D, se utiliza para crear aplicaciones 3D interactivas. Maya nos presenta una arquitectura gráfica por nodos, los elementos en las escenas están basados en nodos, con cada nodo teniendo sus propios atributos. Es así que la representación visual de una escena está basada en una red de nodos interconectados, que dependen de la información de cada uno entre sí.

El software Autodesk Dynamo Studio es un ambiente de programación que permite a los diseñadores crear lógica visual para explorar los diseños conceptuales paramétricos y automatizar tareas. Esta herramienta ayuda a resolver desafíos en menor tiempo al diseñar flujos de trabajo que rigen a la geometría y el

comportamiento de los elementos del diseño. Permite además extender los diseños hacia un flujo de trabajo enfocado en la interoperabilidad para su documentación, fabricación, coordinación, simulación y análisis (Autodesk, 2016).

CATIA o Aplicación Interactiva Asistida por Computadora Tridimensional (Computer Aided Three-dimensional Interactive Application) es una herramienta que fue utilizada por el arquitecto Frank Gehry para diseñar algunos de sus edificios curvilíneos, tales como el Museo Guggenheim de Bilbao. A partir de la experiencia con CATIA, Gehry Technologies, la rama de tecnología de su firma, ha creado su propio software para diseño paramétrico, Digital Project.

El software GenerativeComponents, CAD con capacidades paramétricas, ha sido usado ampliamente en la comunidad arquitectónica, en especial en Londres.

Los usuarios pueden interactuar con el software al modelar dinámicamente la geometría y manipularla directamente, o mediante la aplicación de reglas y definición de relaciones entre los elementos del modelo, o definiendo formas y sistemas complejos a través de algoritmos expresados concisamente.

Marionette es un entorno de programación visual de código abierto, que está enfocado en las industrias de la arquitectura,

ingeniería, construcción y paisajismo. El programa se encuentra incorporado dentro de las versiones de Windows y Mac del software Vectorworks. Esta herramienta les permite a los diseñadores crear algoritmos personalizados que dan lugar a objetos interactivos de carácter paramétrico y flujos de trabajo complejos.

Todo dentro del software consiste de nodos que se unen entre sí. Cada nodo viene predefinido con entradas y salidas.

Modelur es un plug-in paramétrico de diseño urbano para SketchUp, como característica principal le permite al usuario crear masas urbanas conceptuales. El software ofrece herramientas de diseño del entorno construido a través de parámetros urbanos claves, tales como número de pisos y área bruta de construcción en planta de un edificio (Wikipedia, 2016).

### 1.3.1.E. SOFTWARE AUTODESK REVIT

Autodesk Revit es una herramienta BIM muy completa que además de cumplir con todos los requerimientos básicos de un software BIM, presenta grandes ventajas gracias a la facilidad de interoperabilidad con otros softwares BIM, CAD, CAM y de programación visual.

La finalidad de Revit es modelar diseños con precisión, optimizar el rendimiento y colaborar de manera más eficiente.

Autodesk Revit se ha desarrollado específicamente para ser una herramienta BIM, y permite a los profesionales del diseño y la construcción transformar las ideas desde el concepto hasta la terminación con un enfoque coordinado y homogéneo basado en modelos. Autodesk Revit es una sola aplicación que incluye funcionalidad para diseño arquitectónico, construcción, ingeniería MEP y estructural.

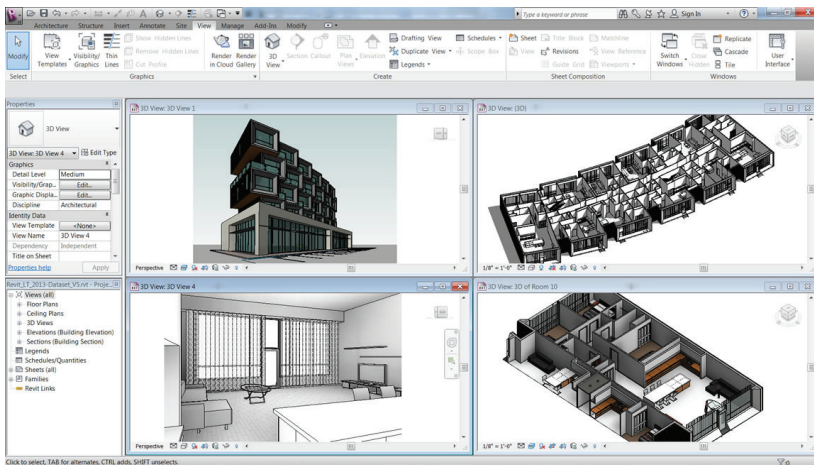
Revit trabaja en base a una base de datos de parámetros, en los que cada tipo de objetos posee parámetros diferentes dependiendo de sus propiedades, esta base de datos sirve para gestionar y coordinar las diferentes decisiones de diseño que se realicen sobre el proyecto. Los cambios realizados en una vista de planta, por ejemplo, se realizan en todo el proyecto, siempre y cuando sean

elementos de modelo, sin necesidad de volver a representar el elemento, Revit presenta las siguientes características:

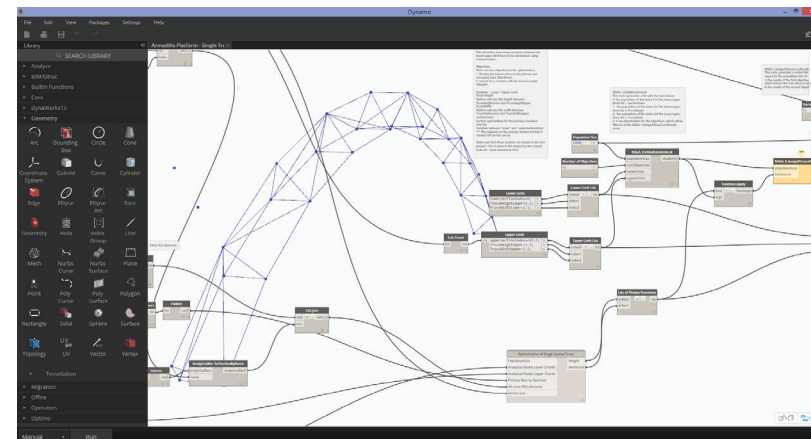
- Componentes de diseño y construcción, el programa cuenta con herramientas para diseñar desde la etapa de conceptualización hasta la entrega de documentos técnicos para la construcción. Esto abarca detalles en muros, pisos, cielos rasos y cubiertas, incluyendo muros cortina. Además, permite realizar un estudio volumétrico mediante masas, cálculo de áreas por pisos y experimentar con texturas, materiales y colores, entre otras aplicaciones.
- Sectoriales, al realizar cualquier cambio en la estructura, orientación y otros detalles que modifiquen la

disposición de elementos frente a la luz, las sombras se ajustan de manera instantánea, permitiendo visualizar el efecto de los cambios en la iluminación, ya sea ésta natural o artificial.

- Perspectivas seccionales, Revit permite analizar el diseño desde todos los ángulos posibles, mediante la utilización de distintas perspectivas y secciones.
- Modelo que integra las diferentes disciplinas, el software posee un conjunto de herramientas para coordinar las distintas disciplinas.
- Produce referencias automáticas de dibujo, estima costos y permite modificar la geometría de los diferentes elementos con tan solo ingresar valores



**Imagen 1.18.** Entorno de trabajo de Autodesk Revit - Captura de pantalla (Autodesk, 2016)



**Imagen 1.19.** Entorno de trabajo de Autodesk Dynamo - Captura de pantalla (Dynamo Primer, 2015)



numéricos, coordina las versiones para que todos los datos, gráficos, detalles y dibujos estén actualizados, entre varias otras funciones orientadas a optimizar los tiempos y mejorar la calidad del proyecto (Hildebrandt, 2015).

- Modelado de terreno y exteriores, permite diseñar mientras se toma en cuenta el contexto exterior. Además, permite planificar los procesos de construcción de forma adecuada.
- Interfaz de usuario multidisciplinaria, distintos equipos pueden trabajar de forma simultánea y el programa coordinará los cambios realizados.
- Presentación y visualización, cuenta con un motor de renderización integrado, el mismo que permite incluir a ventanas y tragaluces en sus cálculos para simular las condiciones de luz natural (Hildebrandt, 2015).

### 1.3.1.F. SOFTWARE AUTODESK DYNAMO

Es un software de diseño computacional algorítmico 3D que funciona de manera independiente o en conjunto con Revit a manera de plug-in, todo lo que se modela dentro del entorno de trabajo adquiere propiedades paramétricas e iterativas. A pesar de pertenecer a la compañía Autodesk el software es "open source" (de código abierto) lo que permite que cualquier persona con conocimiento

de lenguajes de programación pueda inclusive crear su propio contenido para el software. El software utiliza una interfaz de usuario gráfica ideal para la aplicación de la programación gráfica (Huang, 2014).

Al enfocarse en el diseño generativo, Dynamo hace posible la creación de nuevas posibilidades en el diseño paramétrico. Se puede decir que en cierta manera Dynamo es para Revit lo que Grasshopper es para Rhinoceros.

Autodesk desarrolló nuevas características y funcionalidades para el último lanzamiento del software Dynamo y su plug-in para Revit, lo que permite incorporar la programación visual en una plataforma que ya tiene integrado el análisis de diseño y rendimiento (Huang, 2014).

### 1.3.1.G. SOFTWARE RHINOCEROS

Desde la primera versión en 1998, Rhinoceros, o Rhino, se ha convertido en una herramienta de modelado 3D estándar para diseñadores y arquitectos.

Empieza con un boceto, un dibujo, un modelo físico o sólo una idea. Rhino proporciona las herramientas para modelar y documentar los proyectos de diseño con precisión y dejarlos listos para las etapas de renderizado, animación, documentación, ingenierías, análisis y fabricación o construcción.

Rhino puede crear, editar, analizar, documentar, renderizar, animar y traducir curvas NURBS, superficies y sólidos sin límite de complejidad, grado o tamaño. Rhino también es compatible con mallas poligonales y nubes de puntos. Su precisión y flexibilidad permite a los arquitectos explorar y materializar sus ideas sin tener que perder mucho tiempo aprendiendo "CAD" (Visualarq., 2015).

Cualquier geometría creada en Rhino puede exportarse a cortadoras láser, fresadoras o impresoras 3D. Esta opción es la que diferencia a Rhino de las herramientas de modelado 3D generales basadas en polígonos, con las que se pueden crear imágenes, pero sin precisión de fabricación.

La arquitectura abierta de Rhino también permite a los usuarios utilizar Rhino como plataforma de desarrollo: un SDK C++ y una serie de métodos de scripting (Rhino-Script) permiten a los programadores de cualquier nivel de experiencia personalizar y automatizar Rhino, además de ampliar sus posibilidades. Hoy en día, existen cientos de plug-ins comerciales para Rhino con funciones de anidado, creación de terrenos, arquitectura paramétrica, renderizado, animación, CAM, entre otros.

Rhino permite desarrollar sus diseños con rapidez y para comunicar con total fidelidad sus ideas y datos a todas las personas involucradas.



### 1.3.1.H. SOFTWARE GRASSHOPPER

Grasshopper es un editor gráfico de algoritmos estrechamente integrado con las herramientas de modelado 3D de Rhino. A diferencia de Rhino-Script, no requiere conocimientos de programación o de scripting.

Grasshopper: Es un lenguaje de programación visual desarrollado por David Rutten para la compañía Robert McNeel & Associates. Se ejecuta dentro de Rhinoceros 3D. Los programas se crean arrastrando componentes en un lienzo, las salidas a estos componentes se conectan a las entradas de los componentes posteriores (Tedeschi, 2011).

Grasshopper es ahora un robusto editor de programación visual que puede ser extendido por suites o por add-ons externos. Más aún, éste fundamentalmente ha alterado los lujos de trabajo de profesionales a lo largo de múltiples industrias y ha fomentado una comunidad global activa de usuarios. La principal interfaz para el diseño de algoritmos en Grasshopper es el editor basado en nodos. La información va de componente en componente por medio de cables que conectan salidas con entradas. La información puede ser también definida de manera local como una constante, puede ser también importada desde un documento existente de Rhino.

### 1.3.1.I. SOFTWARE Y ARQUITECTURA

Los programas informáticos que permiten trabajar con objetos paramétricos son una nueva herramienta que han ido evolucionando enormemente con el paso de los años. Los elementos que son diseñados adquieren una gran cantidad de información, contienen datos que facilitan el trabajo, y reaccionan ante otros objetos y sus modificaciones.

Los vínculos y relaciones que se establecen entre objetos se ordenan según una estructura en árbol que puede ser modificada. De la misma forma, es posible configurar las partes que conforman un objeto y que se mantienen relacionadas entre sí (García & García, 2010).

El trabajo con estas variables se convierte ahora en el fundamento del proyecto arquitectónico. No se trata ya tanto de imaginar una forma determinada, sino de tener la capacidad de concebir y diseñar las relaciones y reglas que articulan entre sí los objetos, familias y partes componentes de un proyecto arquitectónico.

El software de diseño paramétrico es más que una herramienta de dibujo digital. Además de poder modelar el edificio en 3D y parametrizar las características de sus elementos, permite diseñar formas mucho más eficientes desde el punto de vista acústico, energético y estructural, revisando las modificaciones y diferentes

opciones de manera instantánea (Perez, 2013).

La aparición de programas de diseño y cálculo cada vez más precisos y sofisticados para arquitectura e ingeniería, junto con la evolución tecnológica de los materiales para la construcción, han contribuido a la aparición de nuevas formas edificatorias.

La oferta de software de modelado paramétrico BIM (Building Information Modeling) es muy variada. Softwares como Revit, entre otros, ya se utilizan en la gran mayoría de estudios de arquitectura del mundo.

La fiabilidad de la información del edificio es la característica esencial del BIM y de sus procesos digitales de diseño. Las soluciones BIM que utilizan modeladores del orden de construcción paramétricos proporcionan información del edificio más coordinado, más fiable, de mejor calidad y con mayor coherencia interna que el software CAD de objetos que se ha readaptado a BIM.

### 1.3.2. ALGORITMOS Y ALGORITMOS GENERATIVOS

El término algoritmo es utilizado en la rama de la programación para definir un proceso, una ruta de un diagrama de flujos.

En la Arquitectura no difiere mucho el uso del término, pero la estructura de procesos





que conlleva el realizar un diseño es conformado por algoritmos (Inteligencia Artificial) que son llevados en diferentes rutas dependiendo de las necesidades específicas del cliente, es decir es posible ingresar un cuadro de necesidades y generar un diseño factible (Paiz, 2011).

Los algoritmos en los modelos digitales arquitectónicos podrían estar referenciados en tres aspectos básicos:

- Desarrollo de algoritmos que automatizan tareas de diseño arquitectónico o urbano.
- Creación de modelos de conocimiento de orden arquitectónico computables; es decir que pueden ser resueltos por una computadora en un tiempo y forma razonable.
- Generación automática de variaciones de las características de una arquitectura, en particular de sus estructuras y objetos; en este sentido puede verse como una extensión especial, incluso más compleja, del diseño algorítmico.

Deleuze habla del uso del algoritmo genético en la arquitectura y explica que lo más representativo del mismo es el uso de las matemáticas, su utilización en la edificación y como fuente de concepto para generar ideas; obviando la parte de los cálculos matemáticos, es lo que le da

vida a la obra y la hace posible, es en si la matemática como concepto, como forma de ser y su por qué (Landa, 2001).

Este es un ente que nos ayuda a diseñar objetos en todos sus detalles, es casi como si se aplicara la teoría de la evolución de las especies dentro de la arquitectura, al aplicar algoritmos genéticos, igualmente se irán produciendo generaciones de formas arquitectónicas que el arquitecto deberá seleccionar, recombinar y dirigir.

Más que una arquitectura controlada es aquella arquitectura que logra el balance entre lo real y lo imaginario, la arquitectura que se basa en un principio tangible (algoritmo), para lograr lo imaginado.

Desde cierta perspectiva se entiende al diseño a manera de un código genético, en donde el todo se conforma de pequeños elementos que van llevando información desde los niveles más bajos hasta las instancias superiores, se puede de cierta manera decir que la información evoluciona en su trayecto hasta el nivel jerárquico superior, en donde los datos en un nuevo elemento, pero que contiene la información contenida en el nivel inicial.

Como ejemplo, tenemos dos puntos en el espacio, a partir de estos puntos se crea una línea que los una, este es el resultado del nivel de información superior proveniente de la información de un nivel inferior, si seguimos añadiendo elementos

el resultado irá evolucionando de igual manera. Si a la línea que posee una sola dimensión se le incrementan dimensiones y generamos un plano, por ejemplo, este es el nivel superior con base en esa línea, y así seguirá evolucionando siempre y cuando los datos así lo permitan (Iwamoto, 2009).

Se puede decir que la arquitectura y el diseño siempre es paramétrica, puesto que para llegar a un resultado final siempre es necesario partir de los parámetros que definen las variables para el diseño, pueden ser condicionantes propias del sitio, condiciones climáticas, consideraciones de la función, etc., mientras que no siempre la arquitectura se crea mediante algoritmos generativos, a pesar de estar compuesta de todos estos parámetros.

Se debe entender al algoritmo generativo como una herramienta que facilite los diferentes procesos que están envueltos en el diseño, no simplemente como una herramienta que procese los parámetros ingresados para generar formas aleatorias.

### 1.3.2.A. TRIANGULACIÓN DE DELAUNAY

La triangulación de Delaunay es uno de los tipos de triangulaciones más comunes y útiles que existe. La triangulación de Delaunay de un conjunto específico de puntos debe cumplir un criterio específico, caso contrario no se considera como una triangulación de Delaunay.

AB es un lado de la triangulación de Delaunay, si hay un círculo que pasa a través de A y B de tal manera que todos los otros puntos del conjunto, menos C, dado que A o B no son iguales a C, están fuera del círculo.

De igual manera, todos los triángulos en la triangulación de Delaunay para un conjunto de puntos dado tendrán círculos circunscritos. Es decir, no habrá puntos dentro de la circunferencia circunscrita de ninguno de los triángulos.

Existe solamente una triangulación de Delaunay para cualquier conjunto de puntos dado, este será único, siempre y cuando no existan cuatro puntos que sean co-circulares dentro del conjunto.

### Convex Hull (Casco convexo)

El casco convexo está definido por el polígono convexo de menor área que contenga en su interior todos los puntos de un conjunto de puntos dado (Guillen, s.f.).

### Algoritmo Divide and Conquer (Divide y Vencerás)

El algoritmo divide and Conquer ha sido demostrado como el método de generación más rápido para la triangulación de Delaunay.

Este algoritmo solamente computa la triangulación de Delaunay para el casco

convexo del conjunto de puntos. El primero paso es ordenar todos los puntos en orden ascendente según sus coordenadas en x, si es que dos puntos o más tienen la misma coordenada en x, su orden se determinará por su coordenada en y.

Una vez que los puntos estén ordenados, se debe dividir en el conjunto en mitades hasta obtener subconjuntos que no contengan más de tres puntos. Estos subconjuntos pueden ser triangulados como segmentos en el caso que contengan dos puntos y triángulos cuando contengan tres puntos.

Luego los subconjuntos son unidos con sus mitades anteriores. El resultado será una triangulación que contiene lados-LL (lados presentes anteriormente en la triangulación izquierda), lados-RR (lados presentes en la

triangulación derecha) y lados-LR (lados nuevos que se crearon a partir de la unión de la triangulación izquierda y derecha).

A continuación, se debe insertar el lado-LR base, que es el lado conformado por los puntos inferiores de las dos triangulaciones y que no intersecten ningún lado-LL o lado-RR.

El siguiente paso es definir el próximo lado-LR que se colocará sobre el lado-LR base. Se toman dos puntos que son posibles candidatos para formar el nuevo lado-LR, uno del subconjunto izquierdo y otro del derecho.

Se deben cumplir dos criterios para que el punto escogido sea apropiado para crear el nuevo lado-LR:

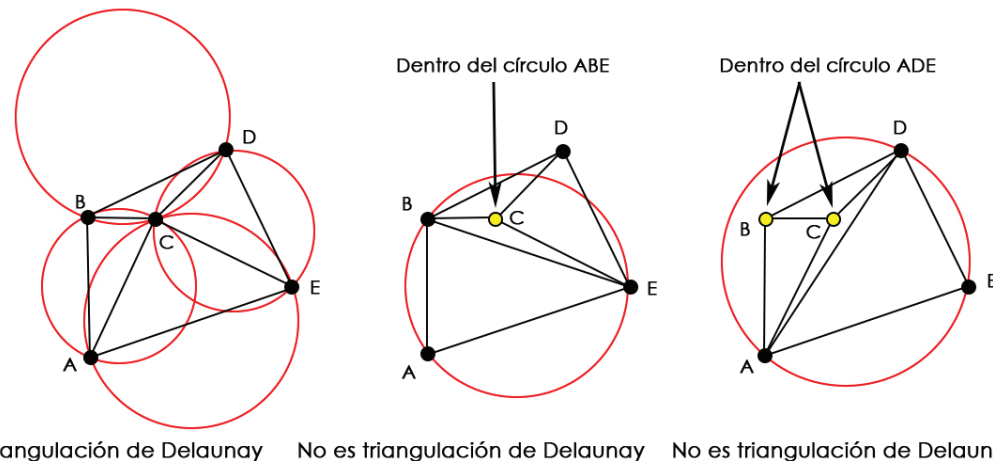


Imagen 1.20. Cumplimiento del criterio para la creación de la triangulación de Delaunay - Imagen digital (Peterson, 1998)



- El ángulo formado por el lado-LR base y el punto examinado debe ser menor a  $180^\circ$ .
- La circunferencia definida por los dos puntos que conforman el lado-LR base y el punto candidato, no debe contener al siguiente punto candidato en su interior.

Si se cumplen los dos criterios, entonces el punto analizado se convierte en el punto definitivo para generar el próximo lado-LR.

Una vez que se determine el nuevo lado-LR se repite todo el procedimiento, con el nuevo lado-LR en lugar del lado-LR base. Se añaden nuevos lados-LR hasta completar toda la triangulación, mientras se realiza la comprobación si alguno de los lados definidos inicialmente intersecta los nuevos lados-LR, se deberán borrar los primeros.

Al terminar de unir las dos mitades originalmente divididas, se completa la triangulación de Delaunay.

### Aplicaciones de la triangulación de Delaunay

A menudo en el diseño gráfico por computadora, problemas de ecuaciones diferenciales y en análisis numérico, es necesario triangular puntos que se encuentran dentro de una región determinada en una malla de alta calidad.

El término, “alta calidad”, hace referencia a la creación de triángulos relativamente uniformes entre sí. Para esto se recurre a la triangulación de Delaunay, posteriormente se aplica un algoritmo que agrega más puntos a la triangulación, definiendo así triángulos cada vez más pequeños y uniformes.

La triangulación de Delaunay es la base para la creación de los diagramas de Voronoi. El diagrama de Voronoi para un conjunto de puntos,  $S$ , es la división del plano en polígonos. Cada uno de los puntos en el conjunto  $S$  se encuentra dentro de un polígono, y cada polígono contiene solamente un punto en su interior. Los polígonos definen el área que está más cerca de un punto determinado que al cualquier otro punto dentro del conjunto  $S$ .

El diagrama de Voronoi es en realidad el inverso de la triangulación de Delaunay. Se puede generar la triangulación de Delaunay a partir de un diagrama de Voronoi al crear los lados que resulten perpendiculares a los lados de los polígonos y viceversa (Peterson, 1998).

### 1.3.2.B. DIAGRAMA DE VORONOI

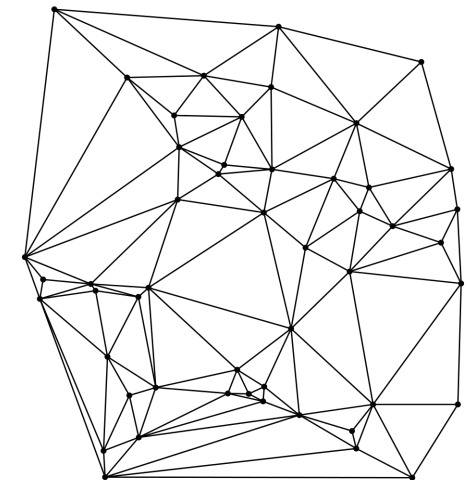
Los matemáticos Dirichlet y Voronoi fueron los primeros en utilizar este término, lo utilizaban para el estudio de formas cuadráticas; los sitios son puntos enteros de una red, y la influencia se mide mediante la distancia euclidiana.

Voronoi fue el primero en considerar la relación de esta estructura, en la cual dos sitios están conectados las regiones de esos sitios tienen un borde común. Luego fue Delaunay quien dedujo que dos sitios están conectados, si sus puntos están sobre una circunferencia en la cual no se encuentra ningún punto del conjunto, esto se llegó a conocer como la triangulación de Delaunay.

### Definiciones y propiedades elementales

La idea del diagrama de Voronoi se basa fundamentalmente en la proximidad.

Suponemos dado un conjunto finito de puntos en el plano:



**Imagen 1.21.** Triangulación de Delaunay para un conjunto de 50 puntos aleatorios - Imagen digital (Wikimedia Commons, 2007)

$$P=\{p_1, \dots, p_n\}$$

Siendo  $n$  mayor o igual a dos, a cada  $p_j$  le asociamos aquellos puntos del plano que están más cerca o igual que de cualquier otro de los  $p_i$  con  $i$  distinto de  $j$ . Todo punto del plano queda así asociado a algún  $p$ , formándose conjuntos que recubren a éste.

Existirán puntos que disten lo mismo de dos elementos de  $P$  y que formarán la frontera de cada región.

Esta teselación recibe el nombre de diagrama de Voronoi en el plano. Cada una de las regiones resultantes se conocen como regiones de Voronoi o polígonos de Voronoi. Los puntos del conjunto reciben el nombre de generadores del diagrama.

Las regiones de Voronoi podrán estar formadas por segmentos de recta, semirrectas o rectas, que llamaremos bordes de la región.

Dentro de los bordes de una región se distinguen aquellos puntos que pertenecen a tres o más regiones, estos tienen el nombre de vértices. Cuando un vértice pertenece a cuatro o más regiones distintas diremos que el diagrama de Voronoi es degenerado. Esto ocurre cuando los generadores correspondientes a cada una de las regiones en las que se encuentra el vértice descansan sobre una misma circunferencia.

Esta configuración es muy inestable, en el sentido de que un pequeño cambio en la posición de cualquiera de los puntos generadores ocasiona la aparición de un

nuevo borde con dos vértices en el lugar donde estaba el vértice original.

Existen varios algoritmos que se pueden implementar para generar el diagrama de Voronoi, que de cierta manera llegan a ser muy parecidos a los que se utilizan para la generación de la triangulación de Delaunay.

### Algoritmo Incremental

Se basa en un diagrama supuesto, construido para  $k$  puntos, a partir de esto se construye el diagrama para  $k+1$ . Este algoritmo emplea un tiempo de  $O(n)$  en la inserción de cada nuevo punto, con una complejidad total de  $O(n^2)$ . A pesar de su complejidad cuadrática, este ha sido el método más popular para construir el diagrama.

### Algoritmo Divide and Conquer (Divide y Vencerás)

El diagrama de Voronoi puede construirse con un algoritmo tipo divide y vencerás en tiempo  $O(n \log n)$ . Esta complejidad es asintóticamente óptima, pero el algoritmo resulta bastante difícil de implementar.

### Algoritmo de Fortune

Hasta mediados de los ochenta, la mayoría de las implementaciones para computar el diagrama de Voronoi usaban el algoritmo incremental cuadrático, admitiendo su

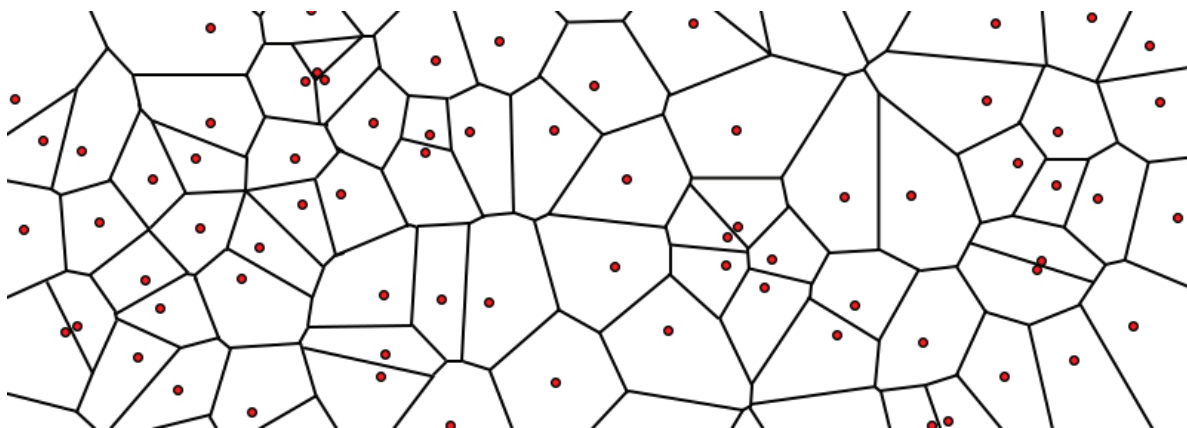


Imagen 1.22. Diagrama de Voronoi para el conjunto de puntos resaltados en rojo - Imagen digital (Wiens, 2013)



mayor lentitud para evitar la complejidad del código divide y vencerás. En 1985 Fortune inventó un inteligente algoritmo de barrido plano que resulta tan simple como el incremental, pero en tiempo  $O(n \log n)$  (Aurenhammer & Klein).

### 1.3.2.C. GRAMÁTICA DE LA FORMA (SHAPE GRAMMAR)

Son sistemas generativos, estos se encuentran definidos y gobernados por una serie de reglas, mediante las cuales se puede capturar, crear y entender los diseños.

Tienen sus raíces en los sistemas de producción de Emil Post, y la gramática generativa de Noam Chomsky.

De igual manera que la gramática se aplica al lenguaje, puede ser aplicada a las formas y geometrías, en lugar de utilizar palabras para formar oraciones, se ocupan directamente formas y las diferentes transformaciones que se pueden aplicar.

Se entienden a las formas como un conjunto finito de líneas, las cuales se modifican mediante la aplicación de una serie de reglas, las mismas que se continuarán aplicando hasta haber completado un diseño, se cumplan una serie de condiciones, o no se pueda aplicar ninguna otra regla.

Para cada etapa del diseño, una o más reglas pueden ser aplicables a la misma forma, lo que produce una gran cantidad

de resultados posibles, para facilitar y limitar la cantidad de resultados que deriven de la aplicación de un conjunto de reglas se puede colocar etiquetas a las partes que conforman la forma (Coutinho, 2013).

Los sistemas generativos hacen referencia a los procesos que, mediante el uso de computadoras, se realizan cálculos basados en una serie de patrones definidos por el usuario.

Los sistemas SG (Shape Grammar) paramétricos les permiten a las formas tener parámetros, que generalmente se expresan en ecuaciones o restricciones, estos sistemas les agregan gran flexibilidad a los resultados, pero al mismo tiempo incrementa la complejidad de los mismos, el número de posibles resultados de estos sistemas se vuelve extremadamente grande (Coutinho, 2013).

Esta lógica puede ser aplicada a la arquitectura, en donde la manera de pensar espacialmente de un arquitecto específico, se puede interpretar mediante la aplicación de una serie de reglas que se presenten de forma clara en sus obras.

Por ejemplo, la relación que existe entre cada uno de los ambientes, dimensiones de los locales, etc. El resultado serán diferentes opciones de distribución, pero todas cumplirán con los criterios de diseño definidos por las reglas aplicadas.

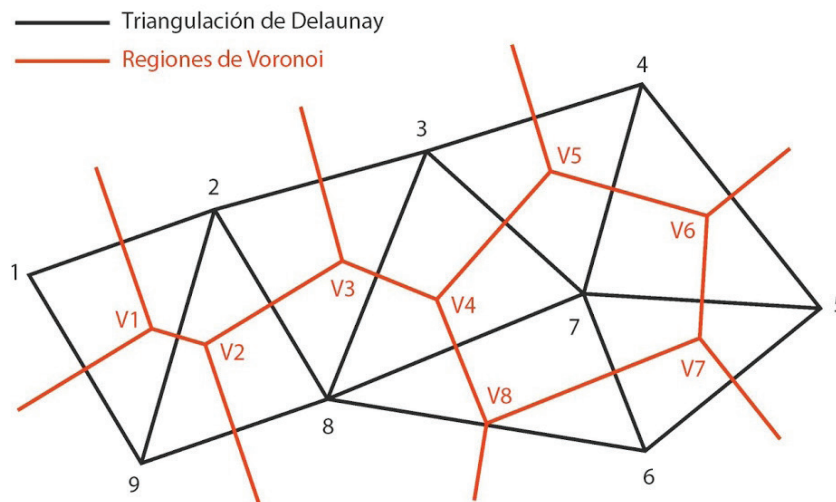


Imagen 1.23. Diagrama de Voronoi y su contraparte la triangulación de Delaunay para un mismo conjunto de puntos - Imagen digital (Soriano, 2013)

### 1.3.3. PROGRAMACIÓN VISUAL Y SU ANATOMÍA

Con frecuencia al diseñar una infraestructura esto implica el establecimiento de relaciones visuales, sistémicas, o geométricas entre las partes de ese diseño. Estas relaciones se desarrollan en diferentes etapas de trabajo, que nos lleva desde el concepto pasando por sus metodología y reglas hasta el resultado final de la obra (Dynamo Primer, 2015).

Tal vez sin saberlo, estamos trabajando de manera algorítmica ya que desde un principio se definen, paso a paso,

las acciones que siguen una lógica básica de entrada, procesamiento y salida. La programación nos permite seguir trabajando de esta manera, pero mediante la formalización de nuestros algoritmos. Los algoritmos pueden generar cosas inesperadas, extravagantes o frías, de hecho, son bastante sencillos.

Vamos a usar un ejemplo tangible como una grulla de origami. Empezamos con un trozo cuadrado de papel (entrada), luego se realizan una serie de pasos ordenados de plegado (acciones), el resultado final tras seguir todos los pasos será una grulla (salida).

El algoritmo es el conjunto abstracto de pasos, que podemos representar en un par de maneras, ya sea textual o gráficamente (Dynamo Primer, 2015).

#### Instrucciones textuales

- Comenzar con un trozo cuadrado de papel, el lado con color hacia arriba. Doblar por la mitad y abrir. Luego doblar por la mitad en el otro sentido.
- Dar vuelta al papel a la cara blanca. Doblar el papel por la mitad, plegar bien y abrir, y luego doblar de nuevo en la otra dirección.
- Usando los pliegues que han hecho, llevar las 3 esquinas superiores del modelo hacia la esquina inferior. Aplanar el modelo.
- Doblar las solapas triangulares superiores hacia el centro y desdoblar.
- Doblar la parte superior del modelo hacia abajo, plegar bien y desdoblar.
- Abrir la solapa superior del modelo, llevándola hacia arriba y presionando los lados del modelo hacia dentro al mismo tiempo. Aplanar, plegar bien.
- Volver a girar el modelo y repetir los pasos 4-6 en el otro lado.

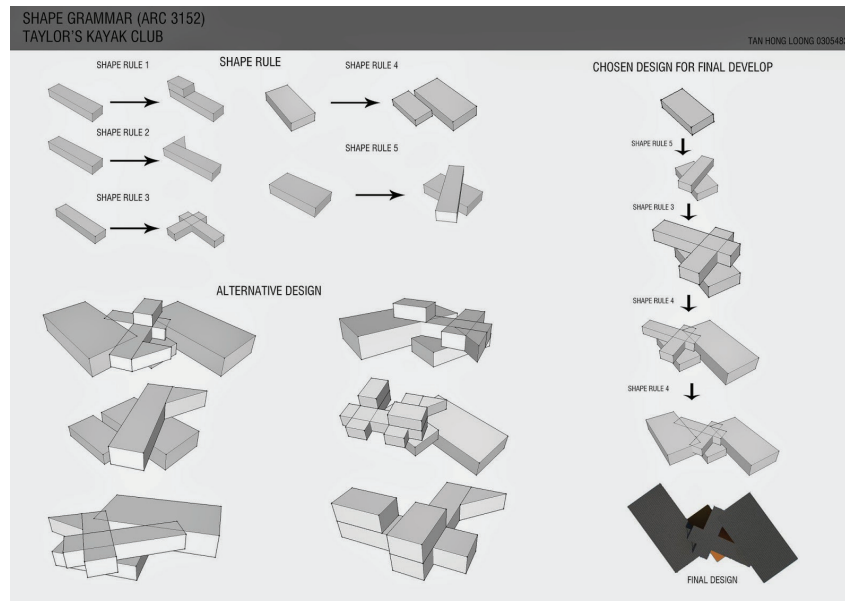


Imagen 1.24. Aplicación de un conjunto de reglas para definir una volumetría - Imagen digital (Tan, 2014)

- Doblar las solapas superiores hacia el centro.
- Repetir en el otro lado.
- Doblar las dos "piernas" del modelo hacia arriba, plegar muy bien, luego desdoblar.
- Realizar un doblez inverso hacia dentro en las "patas" a lo largo de los pliegues que acaba de realizar.
- Realizar un doblez inverso hacia dentro en un lado para hacer la cabeza, luego doblar hacia abajo las alas.
- El resultado final es una grulla.

Las mismas instrucciones se pueden representar de manera gráfica, es decir representar el algoritmo para la creación de la grulla de papel gráficamente.

### Programación

La aplicación de cualquiera de estos conjuntos de instrucciones debe dar lugar a una grulla, esta es la aplicación de un algoritmo. La única diferencia es la forma en la que se lee la formalización de ese conjunto de instrucciones y que nos lleva a la programación, la cual se sintetiza en el ordenador.

La programación es el acto de formalización de la tramitación de una serie de acciones en un programa ejecutable. Sí se ingresan

las instrucciones anteriores para la creación de una grulla en un formato que el ordenador pueda leer y ejecutar, entonces se está programando (Dynamo Primer, 2015).

La clave para librar el primer obstáculo que se encontrará en la programación, es que tenemos que depender de algún tipo de abstracción para comunicarse de manera efectiva con el ordenador. Esto toma la forma de cualquier lenguaje de programación, como Javascript, Python, o C++.

### 1.3.3.A. DEFINICIÓN DE LA PROGRAMACIÓN VISUAL

La programación visual consiste en establecer las instrucciones y relaciones de nuestro programa a través de una interfaz de usuario gráfica o visual, en lugar de ingresar código vinculado por sintaxis.

Como ejemplo a continuación se muestra la diferencia que el mismo algoritmo tiene representado en programación visual y código.

Programación textual (código válido para Autodesk Dynamo)

```
myPoint = Point.ByCoordinates(0.0,0.0,0.0);
x = 5.6;
y = 11.5;
```

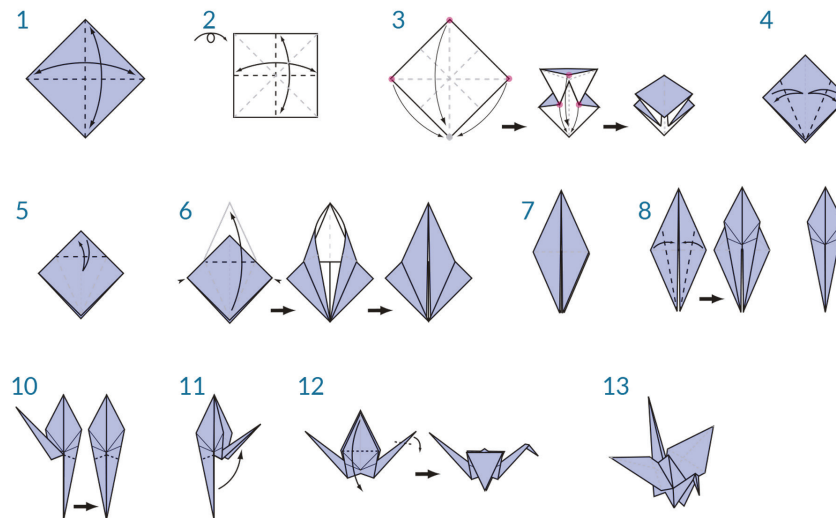


Imagen 1.25. Instrucciones gráficas, algoritmo para crear una grulla de origami - Imagen digital (Dynamo Primer, 2015)

```
attractorPoint = Point.  
ByCoordinates(x,y,0.0);
```

```
dist = myPoint.DistanceTo(attractorPoint);
```

```
myCircle = Circle.  
ByCenterPointRadius(myPoint,dist);
```

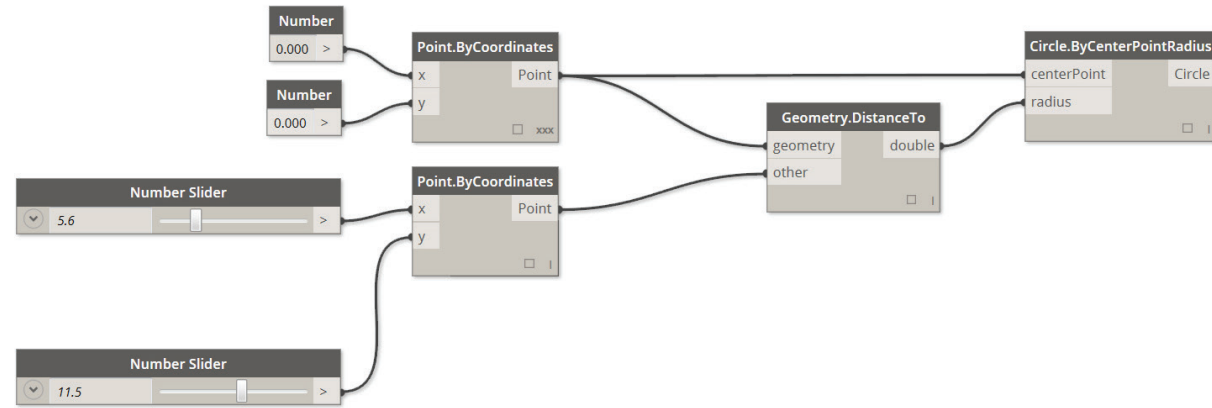
La característica visual de la programación reduce la barrera del ingreso de información y permite a los diseñadores realizar modificaciones sin conocimiento alguno de programación textual. Dynamo cae en el paradigma de la programación visual, pero aun así se puede utilizar la programación textual en la aplicación.

### 1.3.3.B. ANATOMÍA DE LA PROGRAMACIÓN VISUAL

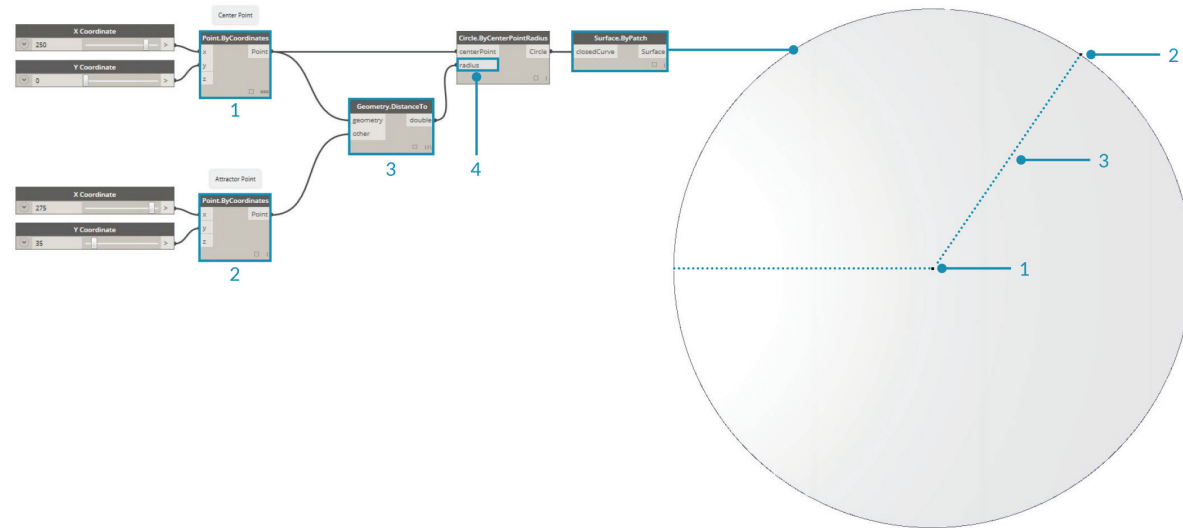
La programación visual nos permite crear algoritmos en un entorno de trabajo gráfico que mediante la conexión de nodos y el uso de cables para especificar el flujo lógico del algoritmo.

#### Anatomía de un nodo

Los nodos son los elementos que se conectan entre sí, para crear un programa o algoritmo. Cada nodo realiza una operación, estas pueden ser tan simples como almacenar un número o pueden ser una acción más compleja como crear o consultar una geometría.



**Imagen 1.26.** Programación visual, algoritmo para generar una circunferencia con base en un punto - Imagen digital (Dynamo Primer, 2015)

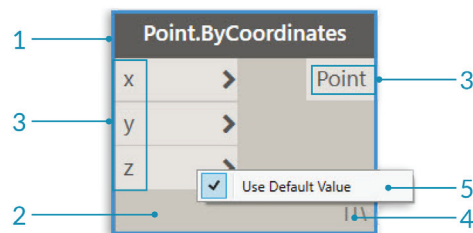


**Imagen 1.27.** Resultado del algoritmo definido anteriormente - Imagen digital (Dynamo Primer, 2015)

La mayoría de los nodos se componen de cinco partes. Aunque hay excepciones, tales como nodos de entrada.

La anatomía de cada nodo puede ser descrita de la siguiente manera:

- Nombre: El nombre del nodo con una convención de Categoría.Nombre.
- Principal: El cuerpo principal del Nodo.
- Puertos (entrada y salida): Los receptores de cables que suministran los datos de entrada al nodo, así como los resultados de las acciones del nodo.
- Ícono de entrelazado: Indica la opción de entrelazado para las listas de información de entrada.
- Valor por defecto: Algunos nodos tienen valores por defecto que se pueden utilizar al hacer clic derecho en un puerto de entrada.



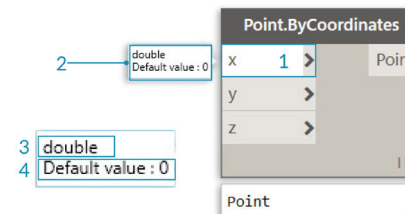
**Imagen 1.28.** Anatomía de un nodo - Imagen digital (Dynamo Primer, 2015)

## Puertos

Las entradas y salidas de los nodos se denominan puertos y actúan como receptores para los cables. Los datos entran en el nodo a través de los puertos de la izquierda y fluyen hacia fuera del nodo después de que ha ejecutado su operación a la derecha. Los puertos esperan recibir datos de un tipo específico. Por ejemplo, la conexión de un número como "2,75" a los puertos de nodo para crear un punto por coordenadas tendrá como resultado con la creación de un punto; sin embargo, si suministramos el texto "rojo" en el mismo puerto, esto dará lugar a un error (Dynamo Primer, 2015).

En los puertos se pueden diferenciar las siguientes partes:

- Etiqueta del puerto
- Información rápida
- Tipo de datos
- Valor por defecto



**Imagen 1.29.** Anatomía de un puerto - Imagen digital (Dynamo Primer, 2015)

## Estado de los nodos

En Dynamo se puede notar el estado de la ejecución del algoritmo mediante la presentación de nodos con diferentes esquemas de color según el estado de cada nodo.

- Activo - Los nodos con un el recuadro del nombre de color gris oscuro están bien conectados y tienen todas sus entradas conectadas exitosamente.
- Inactivo - Los nodos de color gris están inactivos y necesitan estar conectados con cables para ser parte del flujo del programa en el espacio de trabajo activo.
- Estado de error - Los nodos de color rojo indican que están en un estado de error.
- Congelado - Un nodo transparente indica que la opción congelar está activada, lo que impide la ejecución del nodo.
- Seleccionado - Los nodos seleccionados actualmente tienen resaltado en color azul claro en su borde.
- Advertencia - Los nodos de color amarillo indican un estado de advertencia, lo que puede indicar que se han ingresado tipos de información incorrectos para las entradas.



- Pre visualización – Un nodo de color gris oscuro indica que la opción de pre visualizar la geometría está desactivada.

### Cableado

Los cables conectan los nodos entre sí para crear relaciones y establecer el flujo del programa visual. Podemos pensar en

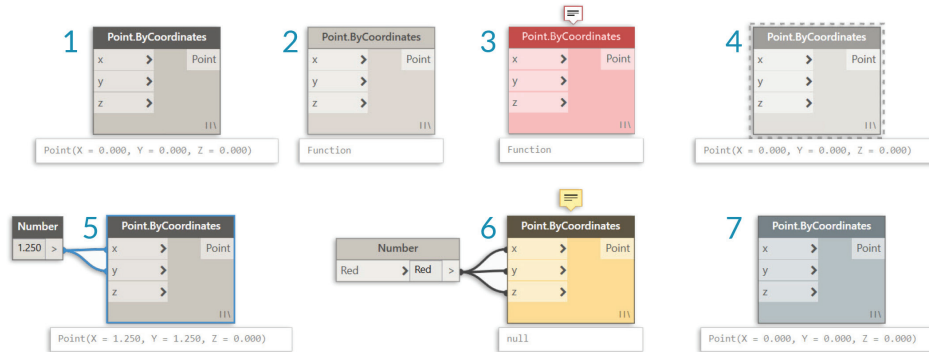


Imagen 1.30. Estado de los nodos en Autodesk Dynamo - Imagen digital (Dynamo Primer, 2015)

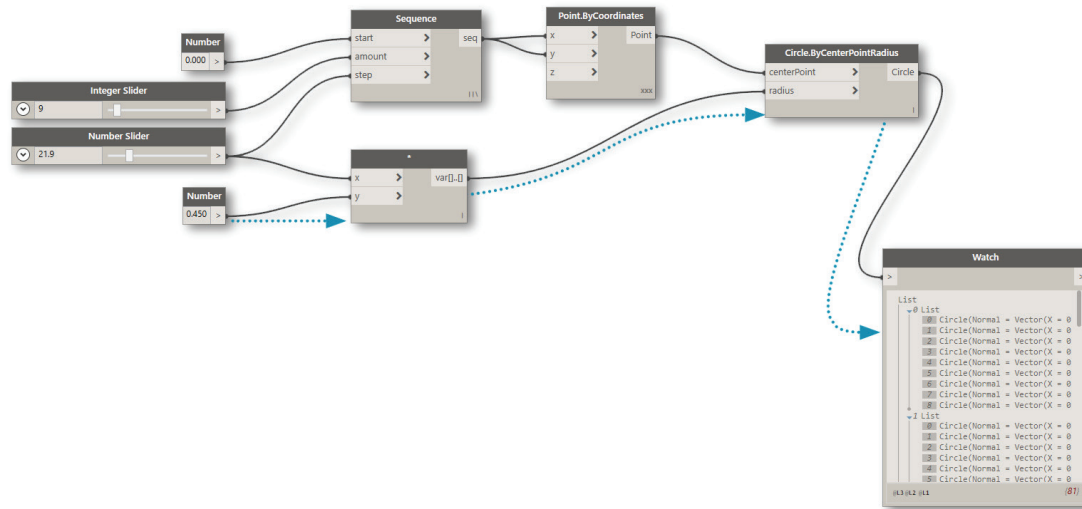


Imagen 1.31. Cableado y flujo de trabajo en Autodesk Dynamo - Imagen digital (Dynamo Primer, 2015)

ellos literalmente como cables eléctricos que transportan los pulsos de datos de un objeto a otro.

Los cables conectan el puerto de salida de un nodo al puerto de entrada de otro nodo. Esta direccionalidad establece el flujo de datos en el programa visual.

Estos nodos se pueden organizar en el espacio de trabajo de cualquier manera, sin embargo, debido a que los puertos de salida están situados en el lado derecho de los nodos y los puertos de entrada están en el lado izquierdo, por lo general, podemos decir que el flujo del programa se mueve de izquierda a derecha.

### La Biblioteca de Dynamo

Esta contiene los nodos que se colocarán en el espacio de trabajo para definir la ejecución de los programas visuales. Los nodos contenidos en la biblioteca, los nodos básicos pre instalados, nodos personalizados y nodos que se instalan desde el administrador de paquetes, están organizados de manera jerárquica por categoría.

La sección de la biblioteca de la interfaz de usuario de Dynamo se compone de bibliotecas organizadas jerárquicamente. Al desplegar el menú de la biblioteca se explora la biblioteca, las categorías de la biblioteca y las subcategorías de las categorías hasta llegar a un nodo.

La jerarquía de cada biblioteca se puede notar en los nombres de los nodos. Típicamente los nombres de los nodos aparecen en el formato Categoría.NombreNodo, con algunas excepciones, principalmente en las categorías de entrada y visualización.

Existen cientos de nodos diferentes, cada uno realiza una acción determinada, pero de manera general los que se utilizarán más frecuentemente son los nodos de entrada, que nos permiten introducir los parámetros en nuestro programa, nodos de visualización, que nos permiten ver los resultados de las acciones de otros nodos y nodos de código, en los cuales se pueden introducir de manera más corta las acciones de otros nodos.

### 1.3.3.c. COMPONENTES DE LA PROGRAMACIÓN

#### Datos

Los datos es la base de los programas. Viaja a través de los cables, entregando información a los diferentes nodos y sus entradas, una vez que son procesados, se convierten en datos de salida.

Los datos son un conjunto de valores que poseen variables cualitativas o cuantitativas. Existen varios tipos de datos, números, variables, caracteres, geometría, o una lista de elementos de datos. Los datos son necesarios para poder ejecutar

las acciones de cualquiera de los nodos.

Existe otro tipo de datos, los nulos ("nulls" en inglés). Estos representan en realidad la ausencia de información y se generan cuando una acción ejecutada por un nodo no produce un resultado válido.

Los esquemas organizativos en los cuales se almacenan los datos se conocen como estructuras de datos, estas son una manera de agrupar datos similares en elementos conocidos como listas.

#### Matemática

La manera más fácil de relacionar la información contenida en los datos de tipo numérico es a través de las diferentes operaciones matemáticas. Se pueden utilizar desde operaciones tan simples como una suma, hasta funciones trigonométricas y fórmulas complejas.

#### Lógica

Los argumentos lógicos, o más específicamente de lógica condicional, nos permiten definir una acción o conjunto de acciones basados en una comprobación. Tras realizarse la comprobación el resultado será un valor que representará verdadero o falso, esto se puede utilizar para controlar el flujo del programa, a estos se le conocen como variables booleanas.

Mientras que las variables numéricas pueden almacenar un gran rango de número diferentes, las variables booleanas solamente pueden almacenar dos valores, verdadero o falso, sí o no, 1 o 0.

El condicional "Sí" ("If" en inglés) es un concepto clave en la programación, ya que nos permite definir que sí se cumplen tales condiciones, se ejecutará una

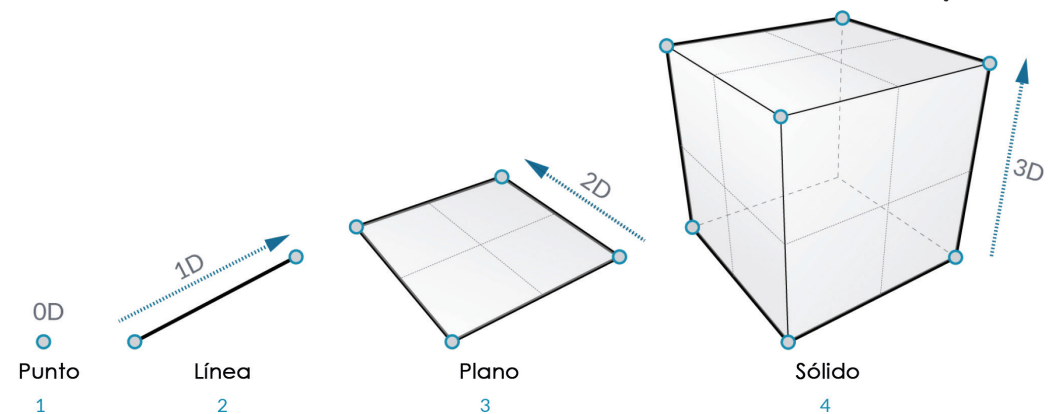


Imagen 1.32. Organización jerárquica de la geometría - Imagen digital (Dynamo Primer, 2015)

instrucción específica, mientras que, si no se cumplen dichas condiciones, se ejecutará algo distinto.

### Cuerdas

Formalmente, una cuerda es una secuencia de caracteres representando una constante literal o algún tipo de variable. Se puede decir que una cuerda es simplemente texto.

### Color

La información que contiene datos de tipo color es muy buena para representar la diferencia entre diferentes resultados en la ejecución de un programa.

Los datos de tipo color se crean utilizando entradas ARGB. Estas corresponden a los canales Alfa, Rojo, Verde y Azul (Alpha, Red, Green y Blue). El canal alfa representa la transparencia del color mientras que los otros canales se utilizan como colores primarios para generar el espectro completo del color.

#### 1.3.3.D. GEOMETRÍA PARA EL DISEÑO COMPUTACIONAL

Los entornos de programación visual permiten definir la manera en la cual son procesados los datos. Los datos pueden ser números o texto, pero también la geometría. La geometría es el tipo de datos que nos permiten crear modelos en

el espacio, así se pueden crear modelos precisos y robustos, automatizar rutinas de diseño y generar iteraciones de diseño mediante algoritmos.

La geometría es fundamentalmente información que puede ser contenida en los datos, y no difiere en gran manera de un número. Los elementos geométricos son descritos como números, relaciones y fórmulas dadas dentro de un sistema de coordenadas espacial.

Está organizada de manera jerárquica, los puntos se unen para formar líneas, líneas se unen para formar superficies y las superficies se unen para formar sólidos.

La dimensión en la que las cosas que se modelan son creadas puede ser utilizada como descriptor de cómo se organiza jerárquicamente la geometría.

Un punto (definido por coordenadas) no tiene dimensión alguna, por lo que solamente son números describiendo cada coordenada.

Una línea (definida por dos puntos) tiene una dimensión, se pueden considerar valores positivos y negativos en el desarrollo de la misma.

Un plano (definido por dos líneas) tiene dos dimensiones, nuevos positivos y negativos.

Una caja (definida por dos planos) tiene tres dimensiones, se puede definir una posición relativa para arriba y abajo.

Existe además una categoría muy diferente de tipo de geometría que es completamente abstracta. Tienen propiedades como la orientación, volumen, o relaciones entre las partes.

Jerarquía según Tipo de Datos							
Tipos Abstractos			Tipos Geométricos				
Define Ubicación + Orientación	Define Posición + Volumen	Define Relaciones	Elementos de modelado				
Vector	Cuadro Delimitador	Topología	Punto	Curva	Superficie	Sólido	Malla
<ul style="list-style-type: none"> <li>Vector</li> <li>Plano</li> <li>Sistema de Coordenadas</li> </ul>	<ul style="list-style-type: none"> <li>Cuadro Delimitador</li> </ul>	<ul style="list-style-type: none"> <li>Vértice</li> <li>Borde</li> <li>Cara</li> </ul>	<ul style="list-style-type: none"> <li>Coordenadas XYZ</li> <li>Coordenadas UV</li> </ul>	<ul style="list-style-type: none"> <li>Línea</li> <li>Polígono</li> <li>Arco</li> <li>Círculo</li> <li>Elipse</li> <li>Curva NURBS</li> <li>Policurva</li> </ul>	<ul style="list-style-type: none"> <li>Superficie NURBS</li> <li>Polisuperficie</li> </ul>	<ul style="list-style-type: none"> <li>Cuboide</li> <li>Esfera</li> <li>Cono</li> <li>Cilindro</li> </ul>	<ul style="list-style-type: none"> <li>Malla</li> </ul>

Imagen 1.33. Cuadro explicativo de la jerarquía de los datos geométricos - Imagen digital (Dynamo Primer, 2015)



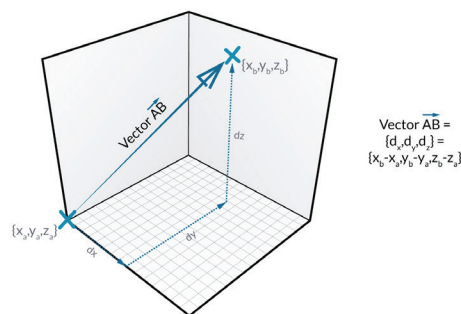
## Vectores, planos y sistemas de coordenadas

Este es el grupo primario de los tipos de geometría abstracta. Nos permiten definir la ubicación, orientación y el contexto espacial para otras geometrías que describen formas.

Los vectores se definen como una cantidad geométrica que describe dirección y magnitud. Son abstractos en el sentido que describen una magnitud, pero no un elemento geométrico físico.

Los planos son elementos abstractos bi-dimensionales. En concepto son planos que se extienden infinitamente en dos direcciones, pero por lo general se los representa como un rectángulo pequeño ubicado cerca de su origen.

Los sistemas de coordenadas son similares a los planos, sí es que son sistemas de



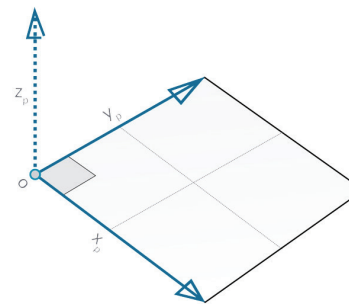
**Imagen 1.34.** Vector dentro de un sistema de coordenadas tri-dimensional - Imagen digital (Dynamo Primer, 2015)

coordenadas euclidianas o XYZ. Sin embargo, existen otros tipos de sistemas de coordenadas tales como los cilíndricos y los esféricos.

## Puntos

Los puntos son la base con la cual toda la geometría es creada, se requieren al menos dos puntos para crear una curva, tres puntos para crear un polígono, etc., al definir la posición, orden y relación entre los puntos (en la función seno, por ejemplo) se puede definir geometría de un orden superior.

Los puntos se definen por uno o más valores llamados coordenadas, la cantidad de valores que se requieren para definir el punto dependen del sistema de coordenadas o el contexto en el que se encuentra. El tipo de punto más común existe en el espacio tri-dimensional y tiene tres coordenadas X, Y, Z.



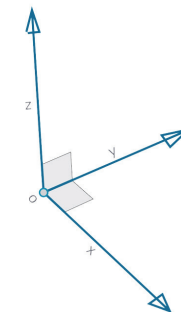
**Imagen 1.35.** Representación de un plano en el espacio - Imagen digital (Dynamo Primer, 2015)

También pueden existir en el espacio bi-dimensional y tienen una notación diferente sí es que existen en un plano, X, Y, o sobre una superficie, U, V.

## Curvas

El término curva hace referencia a todos los tipos diferentes de formas curvas (aún si son rectas), una curva describe a todo punto posible que se puede encontrar al ingresar "t" dentro de una colección de funciones. No importa qué tipo de curva se analice, el parámetro "t" siempre será una propiedad que se puede evaluar. Además, toda curva, sin importar su forma, tiene un punto de inicio y final.

Todas las curvas poseen ciertas propiedades o características que permiten describirlas o analizarlas. Si la distancia entre el punto de inicio y final de la curva es igual a cero, entonces la curva es cerrada. Las curvas tienen una serie de puntos de control, sí



**Imagen 1.36.** Representación del origen de un sistema de coordenadas X, Y, Z - Imagen digital (Dynamo Primer, 2015)

todos los puntos se encuentran en el mismo plano, la curva es planar.

Las líneas son los tipos de curvas más simples, aunque no parezcan curvas, son de hecho curvas, pero sin curvatura. Se definen mediante dos puntos, la forma de la curva se representa entre estos dos puntos, pero matemáticamente se extiende hacia el infinito en ambas direcciones.

Al conectar dos líneas se crea una polilínea, si la polilínea está cerrada, se crea un polígono. Al añadir más complejidad a las funciones paramétricas que definen una forma, se puede llegar a crear arcos, círculos, arcos de elipses o elipses.

Los NURBS (Splines de Base Racional No Uniforme) son representaciones matemáticas que pueden representar de manera precisa cualquier forma a partir de una simple línea bi-dimensional. Tienen gran flexibilidad al ser definidas

por pocos puntos de control, pero poseen una interpolación muy suave gracias a la configuración del grado de la curva.

### Superficies

Una superficie es una forma matemática definida por una función y dos parámetros, en lugar de usar "t" como en las curvas, se utilizan U y V para definir el espacio.

Un dominio en la superficie se define por el rango de parámetros U, V, que denotan un punto tri-dimensional en dicha superficie. Se entiende que las superficies tienen un dominio definido por un mínimo de 0,0 y un máximo de 1,0 tanto para la dirección U como para V.

Se puede entonces definir cualquier punto dentro de una superficie dada según sus coordenadas U, V, y en ciertos casos W.

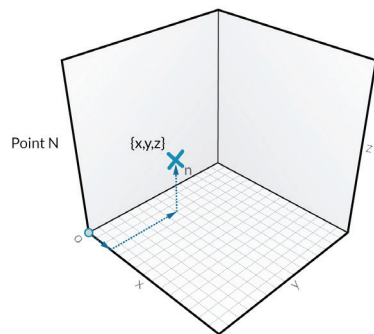
Las superficies NURBS son muy similares a

las curvas NURBS. Se puede pensar en una superficie NURBS como una cuadrícula de curvas NURBS que se extienden en dos direcciones. Las superficies NURBS, independientemente de su aspecto, son cuadrículas rectangulares de puntos de control.

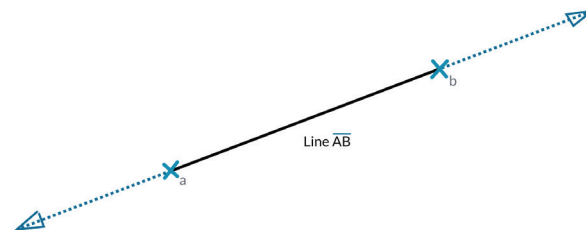
Las polisuperficies son superficies que se han unido a través de sus bordes. Al unir superficies de esta manera es posible crear formas más complejas y detalles en las uniones de las superficies.

### Sólidos

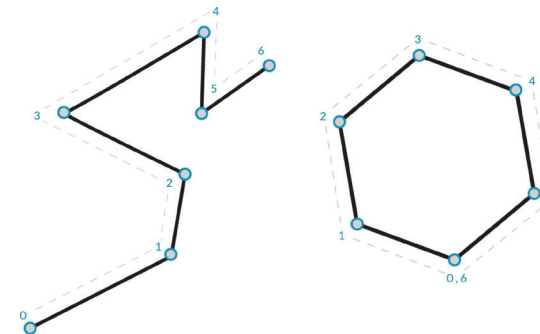
Los sólidos están constituidos por una o más superficies que contienen un volumen en la manera de un límite cerrado. Sin importar el número de superficies que formen el volumen, se debe crear un volumen completamente cerrado para que se pueda considerar como un sólido.



**Imagen 1.37.** Representación de un punto en el espacio - Imagen digital (Dynamo Primer, 2015)



**Imagen 1.38.** Representación de una línea y su proyección hacia el infinito - Imagen digital (Dynamo Primer, 2015)



**Imagen 1.39.** Representación de una polilínea (curva abierta) y un polígono (curva cerrada) - Imagen digital (Dynamo Primer, 2015)

Están formados por tres tipos de elementos: vértices, bordes y caras. Las caras son las superficies que conforman el sólido, los bordes son las curvas que determinan la conexión entre las caras adyacentes y los vértices son los puntos de inicio y fin de esas curvas.

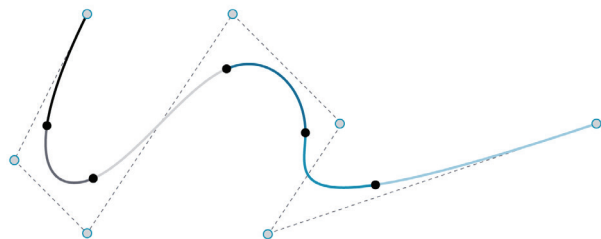
Se pueden combinar dos o más sólidos mediante un método conocido como operaciones booleanas de sólidos.

Las operaciones booleanas principales definir unión, diferencia e intersección en dos o más sólidos.

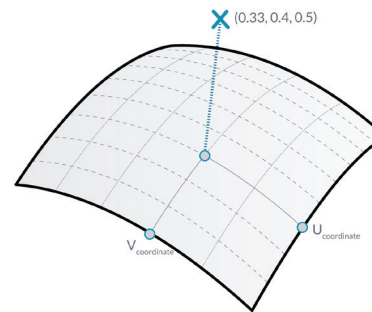
Las operaciones booleanas en sólidos son muy complejas y se pueden demorar una gran cantidad de tiempo para calcular.

## Mallas

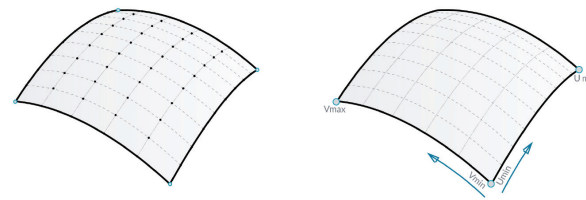
Dentro del campo del modelado computacional, las mallas son de las formas



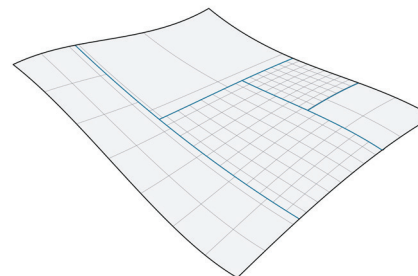
**Imagen 1.40.** Representación de una curva NURBS - Imagen digital (Dynamo Primer, 2015)



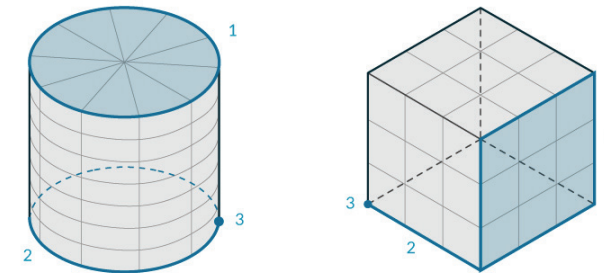
**Imagen 1.41.** Superficie y representación de un punto según sus coordenadas U, V - Imagen digital (Dynamo Primer, 2015)



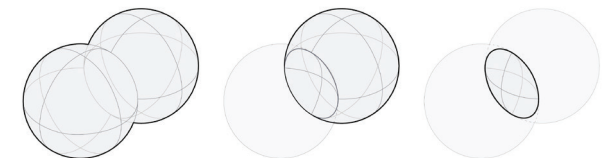
**Imagen 1.42.** Superficies NURBS representadas como una grilla de curvas NURBS - Imagen digital (Dynamo Primer, 2015)



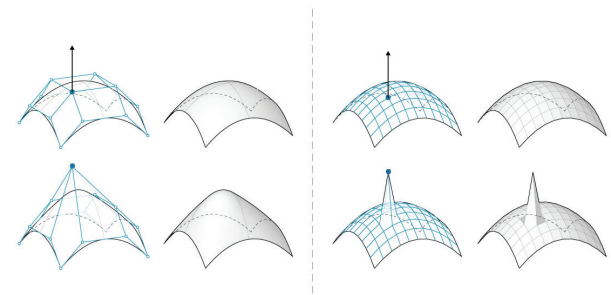
**Imagen 1.43.** Representación de una polisuperficie - Imagen digital (Dynamo Primer, 2015)



**Imagen 1.44.** Representación de dos tipos de sólidos diferentes - Imagen digital (Dynamo Primer, 2015)



**Imagen 1.45.** Diferentes tipos de operaciones booleanas en sólidos - Imagen digital (Dynamo Primer, 2015)



**Imagen 1.46.** Diferencia entre influencia global e influencia local (NURBS vs mallas) - Imagen digital (Dynamo Primer, 2015)



más utilizadas de representar geometrías en 3D. La utilización de mallas puede ser una alternativa flexible y muy liviana al trabajo con NURBS.

Una malla se puede definir como una colección de cuadriláteros y triángulos que representan una superficie o un sólido. Al igual que los sólidos, las mallas se componen de vértices, bordes y caras.

Las mallas están comprendidas por un número discreto de vértices y caras exactamente definidos. El conjunto de vértices no puede ser definido por simples coordenadas UV. La falta de descripción matemática permite que las mallas puedan manejar geometría compleja dentro de una misma malla.

Una diferencia importante entre las mallas y los NURBS es la manera en la cual la modificación de alguno de los elementos locales de la forma afecta la totalidad. El mover un vértice en una malla solo afecta a las caras que se encuentran adyacentes a ese vértice, mientras que en los NURBS crea un cambio mucho más evidente en la geometría.

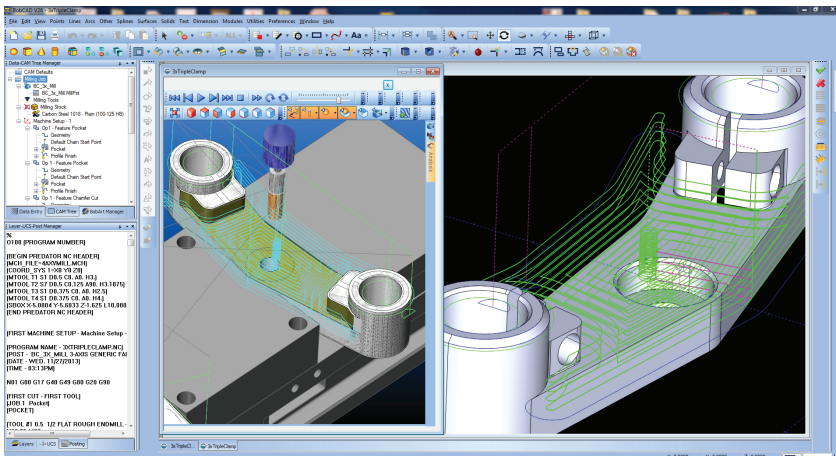
Para entender mejor este concepto se puede comparar una imagen vectorial (compuesta de líneas y curvas) con una imagen rasterizada (compuesta por pixeles individuales). Si se hace un zoom hacia la imagen vectorial, las curvas permanecen nítidas, mientras que el hacer zoom en la imagen rasterizada resulta en observar como cada uno de los pixeles se vuelven más grandes. Los NURBS funcionan como

la imagen vectorial, que mantiene una relación matemática suave y las mallas se parecen a la imagen rasterizada que posee una resolución fija (Dynamo Primer, 2015).

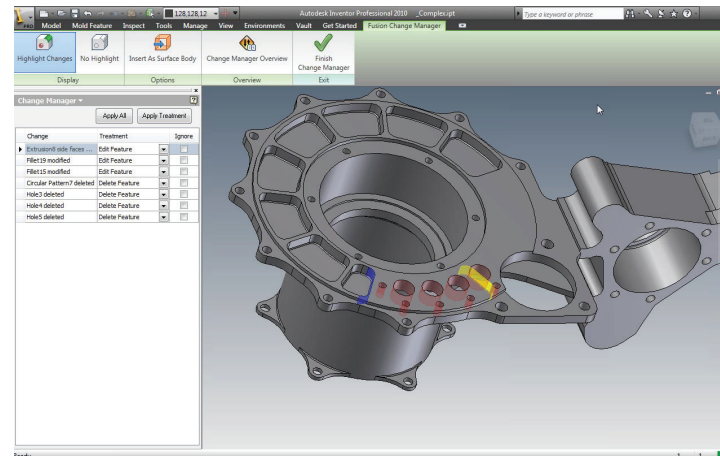
### 1.3.4. EL CAD Y CAM EN EL DISEÑO Y FABRICACIÓN ARQUITECTÓNICA

Los inicios de la tecnología CAD/CAM surgieron por separado y, en ese momento, nadie imaginaba la posible fusión de ambas ni la repercusión económica que dicha fusión tendría. En la industria, el origen del CAD (Computer Aided Design) se le atribuye a Iván Sutherland en 1961 con su trabajo sobre el sketchpad.

Un lápiz dibujaba gráficos en la pantalla y a partir de ahí quedaba digitalizado,



**Imagen 1.47.** Entorno de trabajo de un programa CAM y simulación de un conjunto de instrucciones para fabricación - Captura de pantalla (BobCAD-CAM, 2014)



**Imagen 1.48.** Entorno de trabajo de un programa CAD, diseño de piezas para fabricación - Captura de pantalla (SolidSmack, 2009)



permitiendo hacer zoom, borrar, e incluso guardar como.

Otro ejemplo destacable sería el DAC-1, comercializado en 1964 por General Motors. Este curioso acrónimo (Diseño Argumentado por la Computadora) correspondía a un software que ya podía dibujar los modelos en axonometría, incluso mover y rotar volúmenes.

Por otro lado, la tecnología CAM (Computer Aided Manufacture) comenzó tras la segunda guerra mundial, en 1947, de la mano de John Parson y con la colaboración de la empresa de armamento estadounidense Lockheed y el MIT (Gutierrez, Pérez, Olmo, & Sánchez, 2012).

Tras avances en solitario, la aparición del microprocesador de los 70 y el ordenador personal de los 80 permitieron fusionar ambas tecnologías, introduciendo el bajo coste en la industria y el diseño personal.

La utilización en conjunto del CAD y el CAM está aunando los modos de producción industrial, la concepción y diseño, la fabricación e industria de la edificación, la docencia y los medios de divulgación teórica y crítica.

Este es un modelo teórico-productivo que comparte las NTIC (Nuevas Tecnologías de Información y Comunicación) como herramienta de desarrollo en la arquitectura.

Aparecen por lo tanto cambios sustanciales que afectan a la histórica relación entre representación y producción.

El diseño y la fabricación asistidos por ordenador han alcanzado actualmente un gran nivel de desarrollo e implantación y se han convertido en una necesidad esencial para la supervivencia de las empresas en un mercado cada vez más competitivo. (Gutierrez & et al, 2012).

El Sistema CAD/CAM pretende la automatización total del diseño de nuevos productos, así como en la posterior fabricación de procesos integrados, los cuales se conocen como IAO (Ingeniería Asistida por Ordenador).



**Imagen 1.49.** Maquinaria CNC de 5 ejes, permite gran flexibilidad al crear partes complejas - Fotografía (SHL GROUP, 2016)



**Imagen 1.50.** Maquinaria CNC de tipo Gantry, utilizada para la industria - Fotografía (SHL GROUP, 2016)



El CAD es el sistema que permite la creación y transformación de información gráfica en digitalizada que puede ser procesada en computadoras y almacenada en bases de datos. Se trata de un sistema interactivo, ya que la persona que lo opera puede ir manipulando el concepto inicial y ver el resultado en pantalla (Quiminet, 2012).

En el CAM se utilizan los datos que se han obtenido de la etapa de diseño para planear la producción, fabricación de piezas, el montaje y para el control de calidad de todo el proceso.

El sistema CAM es la programación de robots que operan normalmente en células de fabricación seleccionando y posicionando herramientas y piezas para las máquinas de control numérico.

La planificación de procesos es la tarea clave en para conseguir la automatización deseada, sirviendo de unión entre los procesos de CAD y CAM. El plan de procesos determina de forma detallada la secuencia de pasos de producción requeridos para fabricar y ensamblar, desde el inicio a la finalización del proceso de producción.

#### **1.3.4.A. APLICACIONES DEL SISTEMA CAD/CAM**

Son muchas las disciplinas que pueden beneficiarse al usar este sistema, en especiales aquellas que conciernen a

la integración completa de sistemas. Las industrias más apegadas al funcionamiento del sistema son:

- Mecánica, diseño, análisis y producción.
- Electrónica, diseño y fabricación de circuitos.
- Arquitectura, diseño, documentación y construcción de proyectos.
- Ingeniería, diseño, documentación y construcción de proyectos.
- Construcción, aceleración de procesos.
- Industria textil, diseño y confección.

#### **1.3.4.B. ARQUITECTURA Y EL SISTEMA CAD/CAM**

El uso en conjunto de software CAD y CAM ofrece la posibilidad de agilizar y facilitar extraordinariamente el diseño de las variables técnicas, los planos o prototipos, en la arquitectura moderna.

En un sentido amplio, se puede entender el CAD/CAM como la aplicación de la informática al proceso de diseño y planeamiento de producción, es decir la fabricación del modelo.

En las diferentes etapas del proyecto

arquitectónico se deben mantener organizados los diferentes procesos técnicos, de esta manera se provee una fuente de referencia para los analistas, diseñadores y constructores (Rodriguez, 2014).

Los fundamentos de los sistemas de diseño y fabricación asistidos por ordenador son muy amplios, abarcando múltiples y diversas disciplinas.

El software CAD/CAM permite a los usuarios utilizar herramientas de modelado geométrico, diferentes técnicas de visualización que son esenciales para la representación de los resultados de los algoritmos utilizados, técnicas de interacción gráfica que permite al usuario ingresar información dentro del programa (WordPress.com, 2013).

Uno de los aspectos más importantes de una aplicación CAD/CAM es su interfaz. Del diseño de la misma depende en gran medida la eficiencia de la herramienta, los softwares CAD/CAM además utilizan una base de datos para almacenar toda la información del modelo, desde los datos de diseño, los resultados de los análisis que se realicen hasta la información de fabricación.

El modelado en tres dimensiones es la puerta de entrada a un ambiente CAD/CAM completo. A pesar de que los sistemas 3D no son necesariamente ocupados para

todos los ambientes de diseño, ingeniería y manufactura, muchos de los sistemas CAD/CAM tri-dimensionales pueden replicar las funciones de sistemas 2D si así se requiere (Moreno, 2012).

El modelado mediante sólidos dentro del ambiente CAD/CAM permite agregar un factor determinante para automatizar el diseño a través del proceso de manufactura, esta herramienta permite almacenar información precisa sobre piezas dadas.

La manufactura integrada por computador (CAM por sus siglas en inglés) reconoce que los diferentes pasos en el desarrollo de productos manufacturados están interrelacionados y pueden ser ajustados de manera más eficiente y efectiva con el uso de computadoras.

Para conseguir el producto final en una secuencia CAD/CAM se deben seguir una serie de pasos, primero en la etapa de diseño se utilizará el software CAD para definir el diseño, luego se definirán los pasos para la manufacturación del diseño mediante un software CAM, después se realiza una simulación de los pasos de manufacturación para poder detectar cualquier posible error y corregirlo, tras comprobar que los pasos se ejecuten adecuadamente se genera el código o programa CNC, se ingresa el código dentro del controlador de una máquina CNC para que se realice la interpretación del mismo

y finalmente se hace el maquinado que es la etapa final de producción del diseño inicial.

La implementación de este sistema por parte de algunas empresas busca, por un lado, aumentar la productividad y, por otro, mejorar la calidad de los productos, lo mismo se puede decir de la utilización del CAM dentro de la arquitectura.

#### 1.3.4.c. EL SISTEMA DEL SOFTWARE BIM

El BIM (Building Information Modeling) es un proceso basado en modelos inteligentes que proporciona información para la creación y gestión de proyectos de construcción de manera más rápida, más económica y con un menor impacto ambiental.

El BIM se entiende como el proceso de generación y gestión de datos del edificio, utilizando software dinámico de modelado en tres dimensiones, para disminuir la pérdida de tiempo y recursos en el diseño y la construcción (Lee & Eastman, 2006).

Este proceso produce el modelo de informático del edificio (también abreviado BIM), que abarca la geometría del edificio, las relaciones espaciales, la información geográfica, así como las cantidades y las propiedades de sus componentes.

El BIM puede ser utilizado para ilustrar el proceso completo de edificación, de mantenimiento e incluso de demolición. Las cantidades de materiales y las propiedades compartidas pueden ser extraídas fácilmente. Además, se pueden aislar y definir los ámbitos laborales, detalles de componentes y secuencias de actividades de construcción.

Los programas CAD básicamente se enfocan en dibujar y representar los elementos en 2D y 3D, sin considerar la necesidad de que los elementos puedan almacenar información fuera de sus propiedades geométricas básicas, los programas BIM están desarrollados con la incorporación de estos en mente.



Imagen 1.51. Softwares y formatos BIM más utilizados en el medio - Imagen digital (Fabricantes BIM, 2016)



El software BIM presenta las siguientes características:

- Cuando hay varios usuarios trabajando en un mismo proyecto, la coordinación no es tan complicada como con el software CAD, el BIM permite identificar las interferencias que se presenten de forma inmediata.
- Aumento en la productividad, menos horas hombre que se traduce directamente en menores costos.
- El sistema permite dedicar más tiempo al diseño ya que se reduce el tiempo en que hay que pasar los bosquejos iniciales a CAD, además, este sistema facilita el diseñar todos los detalles.
- La base de datos del software BIM permite controlar la información de forma óptima, ya que esta se convierte en la fuente central para toda la información del proyecto.
- Facilita la interacción entre el cliente y el diseñador, ya que de una manera muy sencilla se puede mostrar el avance del diseño en 3D (Assael, 2006).

El modelo BIM se caracteriza por poseer equivalentes virtuales de los elementos constructivos y piezas que se utilizarán para construir el proyecto. Estos elementos tienen todas las características físicas y

lógicas de sus componentes reales. Estos elementos inteligentes son el prototipo digital de los elementos físicos del edificio, como son los muros, pilares, ventanas, puertas, escaleras, etc.

La utilización de la tecnología BIM permite diseñar, definir y monitorizar los diferentes aspectos del diseño de una edificación, desde el momento de diseño conceptual hasta la etapa de funcionamiento del edificio, estos tipos de software han buscado integrar todas las ramas que intervienen en el proceso de diseño y construcción de un edificio, lo que las convierte en una herramienta necesaria en los momentos actuales.

### 1.3.5. PROGRAMAS SELECCIONADOS PARA REALIZAR EL PROYECTO

Durante la realización de este estudio han destacado principalmente dos softwares que se encuentran orientados a la creación de contenido paramétrico mediante la utilización de la programación visual, estos son Grasshopper y Dynamo, al ser prácticamente herramientas de creación de algoritmos y relaciones necesitan trabajar con otro software más potente para poder representar los resultados de los algoritmos, siendo así, Grasshopper trabaja en conjunto con Rhinoceros, mientras que Dynamo trabaja con Revit.

Los dos son plug-ins que se ejecutan dentro del entorno de trabajo de otro software.

Siendo así, se ha hecho un análisis comparativo sobre los softwares anteriores en relación a su función, es decir Grasshopper con Dynamo y Rhinoceros con Revit.

#### 1.3.5.A. SOFTWARE REVIT

##### Ventajas

- Interfaz de usuario muy ordenada, con facilidad de acceso a las diferentes herramientas.
- Instalación rápida y sencilla, los datos necesarios se descargan directamente de internet.
- Software flexible, que permite la realización de trabajos de toda escala.
- Incorporación de herramientas orientadas a diferentes ramas de la construcción, como por ejemplo ingeniería.
- Todos los elementos modificados se actualizan en tiempo real.
- Se puede abarcar todas las fases del proyecto, desde sus inicios, hasta la etapa de funcionamiento.
- Al ser un programa que trabaja en tres dimensiones permite el cálculo de volúmenes de obra.





- Facilidad de acceso al software completo sin limitaciones por parte de estudiantes.
- Rapidez de trabajo, gracias a la generación automática de elementos en todas las vistas.
- Permite la colaboración de varios equipos de profesionales en el mismo archivo y al mismo tiempo.
- Gran cantidad de herramientas con funcionalidad en la nube.
- Elementos que pueden contener gran cantidad de información y parámetros.

#### Desventajas

- Se dificulta el dibujo superficies y volúmenes en 3D complejos.
- Las herramientas de dibujo son bastante estrictas respecto a los métodos de creación de elementos.
- Las herramientas de renderizado no son competitivas frente a otros softwares especializados.
- Debido a la gran cantidad de información que contienen los elementos, algunos archivos pueden tener tamaños muy grandes, lo cual puede dificultar el trabajo en computadoras no especializadas.

- Algunas herramientas, sobre todo las dedicadas al modelado de entornos de trabajo no son lo suficientemente adecuadas.
- Trabaja con un grupo limitado de extensiones de programa, los tipos de archivos que acepta son muy pocos.
- El costo del programa completo para profesionales es bastante elevado.

#### 1.3.5.B. SOFTWARE RHINOCEROS

##### Ventajas

- Es de fácil uso, las herramientas básicas son bastante intuitivas.
- Herramientas muy robustas para el modelado 3D.
- Posee gran precisión, lo que lo favorece para trabajos enfocados en análisis, prototipado y fabricación.
- Puede trabajar con gran cantidad de formatos y extensiones, es compatible directamente con la mayoría de programas de diseño en el mercado.
- No requiere de hardware especializado para poder ejecutar el programa.
- Gran cantidad de plug-ins que mejoran y potencializan las capacidades del software.

- Permite establecer procesos de fabricación.
- Contiene herramientas destinadas al análisis de las diferentes etapas de diseño.
- Permite gran libertad en el modelado de formas, capacidad de crear formas orgánicas.

##### Desventajas

- La interfaz de usuario puede ser algo confusa al inicio.
- Se encuentra más orientado al diseño de joyería y náutico.
- No posee una herramienta que permita la creación de planos técnicos.
- Dificultad en la edición de los elementos cuando estos se han vuelto muy complejos.
- No posee herramientas enfocadas en la simulación.
- Es necesario obtener el software completo para profesionales para poder utilizarlo pasado los 90 días de prueba.
- Tiene una curva de aprendizaje muy grande para llegar a explotar todas las capacidades del software.



### 1.3.5.C. SOFTWARE DYNAMO

#### Ventajas

- Es bastante intuitivo, las herramientas vienen agrupadas según las acciones que realizan.
- Permite una rápida iteración de diseños, a partir de un algoritmo.
- Puede extraer datos de archivos externos y utilizarlos como valores dentro de los elementos del diseño.
- La base de este software es de código abierto, lo que permite que todos puedan crear contenido personalizado y compartirlo con la comunidad.
- Curva de aprendizaje muy corta, el flujo de trabajo es bastante directo.
- Los estudiantes pueden acceder a la versión completa del software de manera libre.
- Capacidad de automatizar procesos repetitivos dentro de Revit.
- Le agrega una capacidad paramétrica enorme a Revit, al igual que de modelado de formas y superficies complejas.
- Es muy fuerte en el trabajo con listas.

#### Desventajas

- Es un software relativamente nuevo y que sigue en desarrollo.
- Carece de algunas herramientas o procesos que se pueden pensar básicas, que deben ser creadas manualmente.
- Al ser un software nuevo, hay poco nivel de información en la red.

### 1.3.5.D. SOFTWARE GRASSHOPPER

#### Ventajas

- Es fácil de aprender, gracias a la gran cantidad de información que hay en la web.
- Entorno de trabajo basado en la programación visual.
- Permite la automatización de procesos.
- Permite la creación de geometrías con un muy alto grado de complejidad.
- Es un software muy preciso y exacto, que tiene varios años de desarrollo.

#### Desventajas

- Depende del software Rhinoceros para ejecutarse.

- Si bien es fácil de usar, para poder utilizarlo de manera muy eficiente y demostrar el poder que posee como herramienta, se requiere de grandes conocimientos.

### 1.3.5.E. VALORACIÓN DE LOS SOFTWARES BIM

Se han creado 8 criterios de valoración según los cuales se determinará el software más adecuado para el desarrollo del trabajo, cabe recalcar que esta valoración se ha determinado únicamente por los conocimientos y experiencias de los autores sobre los softwares analizados.

Los campos a evaluar son:

- Facilidad de uso.
- Facilidad de configuración.
- Facilidad de administración.
- Herramientas de modelado 3D.
- Herramientas de dibujo.
- Herramientas de edición.
- Interoperabilidad entre archivos.
- Conocimiento previo del software.

Los valores asignados a cada software se describen en la siguiente tabla:



Software	Facilidad de uso	Facilidad de configuración	Facilidad de administración	Herramientas de modelado 3D	Herramientas de dibujo	Herramientas de edición	Interoperabilidad entre archivos	Conocimiento previo del software	TOTAL
Autodesk Revit	4,0 / 5,0	4,0 / 5,0	4,5 / 5,0	3,5 / 5,0	3,5 / 5,0	4,0 / 5,0	3,0 / 5,0	5,0 / 5,0	31,5 / 40,0
McNeel Rhinoceros	4,0 / 5,0	4,5 / 5,0	4,0 / 5,0	5,0 / 5,0	4,0 / 5,0	4,0 / 5,0	5,0 / 5,0	0,0 / 5,0	30,5 / 40,0

**Imagen 1.52.** Tabla de valoración para los softwares BIM - Imagen digital (Elaboración propia, 2016)

Software	Facilidad de uso	Herramientas de modelado 3D	Herramientas de dibujo	Herramientas de edición	Interoperabilidad entre archivos	Conocimiento previo del software	TOTAL
Autodesk Dynamo	5,0 / 5,0	4,0 / 5,0	4,0 / 5,0	4,0 / 5,0	4,0 / 5,0	3,0 / 5,0	24,0 / 30,0
Grasshopper	3,0 / 5,0	5,0 / 5,0	4,0 / 5,0	4,0 / 5,0	5,0 / 5,0	0,0 / 5,0	21,0 / 30,0

**Imagen 1.53.** Tabla de valoración para los softwares paramétricos - Imagen digital (Elaboración propia, 2016)

El software Revit obtiene una valoración de 31,5 sobre 40 puntos posibles, mientras que Rhinoceros obtiene una valoración de 30,5 sobre 40.

#### 1.3.5.F. VALORACIÓN DE LOS SOFTWARES PARAMÉTRICOS

Para determinar el software a utilizar se han establecido 6 criterios para la valoración:

- Facilidad de uso.
- Herramientas de modelado 3D.
- Herramientas de dibujo.
- Herramientas de edición.
- Interoperabilidad entre archivos.
- Conocimiento previo del software.

Los valores asignados para el software Dynamo le dan un total de 24 sobre 30 puntos, mientras que para Grasshopper el total es de 21 sobre 30.

Al analizar los softwares disponibles para la creación de diseño paramétrico mediante la utilización de programación visual, se ha establecido por los autores, que el software que se utilizará es Autodesk Dynamo, que trabaja en conjunto con Autodesk Revit, en conjunto reciben una valoración de 55,5 sobre 70.

A pesar de que los softwares Rhinoceros y Grasshopper son los más populares en el medio, han destacado otros dos softwares, que al igual que los anteriores, trabajan en conjunto para la creación del diseño paramétricos, estos son Autodesk Revit y Autodesk Dynamo, que al pertenecer a la misma compañía fueron creados

para que el uno complementa al otro directamente. Dada la gran versatilidad del software Revit como herramienta BIM, además del conocimiento del manejo con el que contamos los autores del presente proyecto, y el gran parecido de Dynamo con el plug-in Grasshopper para Rhinoceros, se ha optado por utilizar Autodesk Revit en conjunto con Autodesk Dynamo para el desarrollo del trabajo.

### 1.4. CASOS DE ESTUDIO

#### 1.4.1. PROYECTO DIGFABMTY 1.0

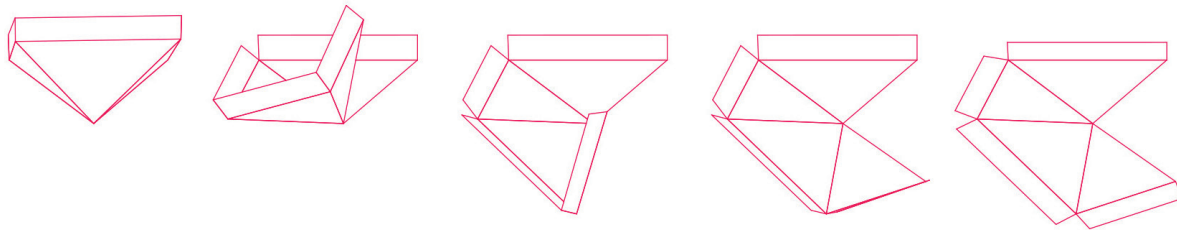
Este proyecto fue realizado por estudiantes de la Escuela de Arquitectura, Arte y Diseño del Tecnológico de Monterrey, y tiene como objetivo la exploración de las técnicas de fabricación digital, al mismo tiempo que se buscó determinar la factibilidad de la construcción y generación de este tipo de estructuras.

El pabellón, que recibe el nombre de DIGFABMTY 1.0, utiliza el método de creación mediante algoritmos, que definen la forma básica del elemento y permiten la posterior manufactura de piezas y componente para la puesta en obra.

Se puede notar la intencionalidad de los diseñadores para cambiar la altura del elemento a lo largo de su desarrollo para permitir que esto genere juegos interesantes con el ingreso de luz desde el exterior,

además de estar compuesto de un módulo básico que se repite iterativamente y se ajusta al espacio disponible.

Se utilizaron materiales con un costo relativamente bajo, sin embargo, el costo total de toda la estructura es comparativamente más alto que un espacio de igual dimensión en técnicas tradicionales, esto puede ser debido al alto costo y baja difusión de técnicas para este tipo de manufactura (máquinas CNC).



**Imagen 1.54.** Proceso de doblado del módulo básico que conforma el pabellón - Imagen digital (ArchDaily, 2015)



**Imagen 1.55.** Resultado final del ensamblaje del pabellón - Fotografía (ArchDaily, 2015)

El proyecto se desarrolló con Grasshopper y Rhinoceros, utilizando Coroplast y PVC como materiales.

#### 1.4.2. LEVANTAMIENTO DEL CENTRO DE LA CIUDAD DE MÁLAGA

A continuación, se describe un caso práctico sobre cómo se ha levantado el entorno del centro de Málaga.

El trabajo se ha desarrollado con el uso de las herramientas Autodesk Revit y Autodesk Dynamo, dentro de Revit se creó un nuevo proyecto en el cual se incorporará el entorno.

Dentro de un navegador web se accedió a la herramienta online OpenStreetMap y se ha seleccionado la parte de la ciudad con la que se desea trabajar, para este caso específico, el centro de la ciudad de Málaga.

Posteriormente se exportó el fichero en formato \*.osm. Dentro de Dynamo, se instaló los paquetes de topografía "ELK" y osm.

Dentro de los paquetes de nodos personalizados que se han instalado se encuentran dos especialmente que nos permitirán generar una volumetría a partir del archivo en formato \*.osm que hemos guardado.



El nodo OSM.Location nos permite definir la ubicación del archivo que vamos a utilizar dentro del computador, es decir la ruta del archivo, mientras que el nodo OSM.OSMData nos permite definir qué tipo de información queremos extraer del archivo \*.osm.

Los otros nodos utilizados vienen incorporados en Dynamo, PolyCurve. ByPoints que genera una policurva a partir de un conjunto de puntos, Levels que permite seleccionar un nivel existente dentro del proyecto de Revit, Floor Types que permite seleccionar el tipo de piso a crear según los tipos existentes en el proyecto de Revit, PolyCurve.CloseWithLine que cierra una policurva añadiendo una línea entre el punto de inicio y el punto final de la policurva, y finalmente Floor.

ByOutlineTypeAndLevel que genera un piso en el entorno de trabajo de Revit a partir del contorno y nivel en el que se desea crearlo.

Tras la aplicación de estos nodos el resultado serán los contornos de las edificaciones ya levantadas en el proyecto de Revit. Se pueden agregar diferentes tipos de pisos que se utilicen de acuerdo a la altura de cada edificación para esta manera asociarlos con los volúmenes creados.

## 1.5. MARCO METODOLÓGICO

### 1.5.1. TIPO Y DISEÑO DE LA INVESTIGACIÓN

El presente trabajo se ha enmarcado en una investigación de tipo exploratoria,

para el desarrollo de la investigación será necesario consultar varias fuentes que hagan referencia sobre este tema tan poco conocido en nuestro medio, además, la investigación es de diseño experimental, puesto que todos los resultados que deriven de la misma serán producto de la experimentación y exploración de los resultados y las varias iteraciones que genere el proyecto.

### 1.5.2. POBLACIÓN Y MUESTRA PARA LA INVESTIGACIÓN

Como población base para la investigación se ha definido todo el universo de lo que se conoce como diseño paramétrico, más específicamente el diseño paramétrico que se genera mediante el uso de algoritmos y software CAD. Una muestra

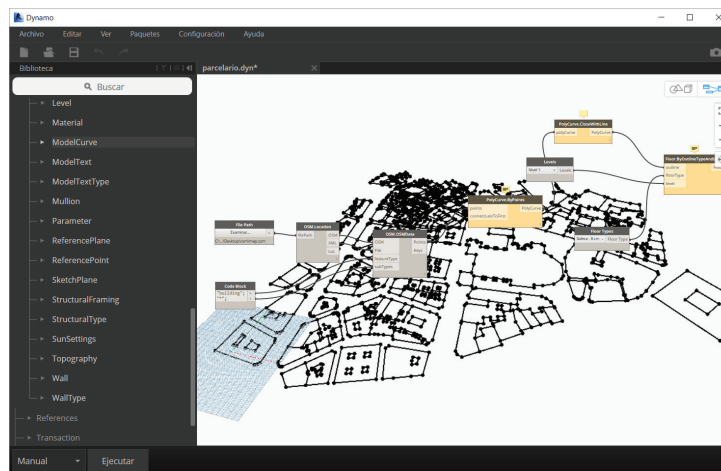


Imagen 1.56. Algoritmo utilizado para obtener el contorno de las edificaciones del centro de la ciudad de Málaga - Captura de pantalla (GARQUITECTOS, 2016)

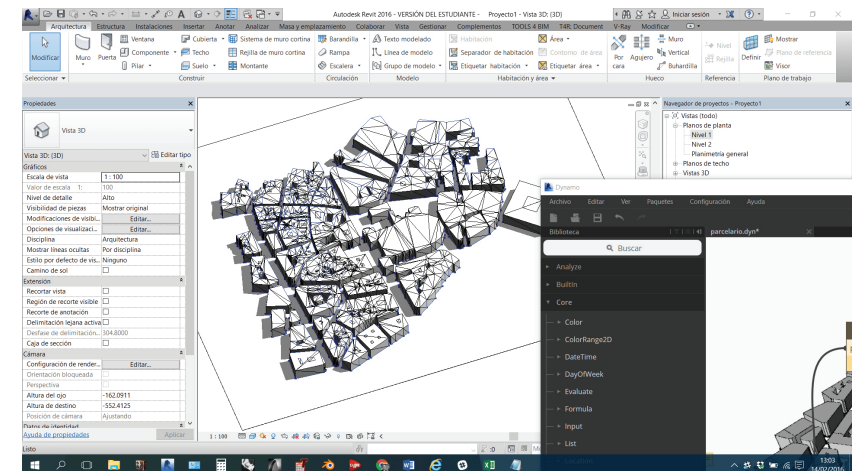


Imagen 1.57. Resultado de la aplicación del programa visual dentro del entorno de trabajo de Revit - Captura de pantalla (GARQUITECTOS, 2016)



representativa de este tipo de diseño es aquella que comprende espacios habitables y pabellones.

### 1.5.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Para el desarrollo del trabajo de titulación se han utilizado las siguientes técnicas para la recolección y recopilación de información:

- Acopio de textos, tesis, revistas, artículos periodísticos, sitios web.
- Consulta a manuales y tutoriales sobre la utilización del software escogido para la realización del trabajo.
- Elaboración de información primaria y obtención de información secundaria.
- Observación directa y experimentación con el software propuesto.
- Análisis del software elegido para la realización del proyecto.
- Análisis e interpretación de la información obtenida.

### 1.5.4. TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS

Los métodos utilizados para procesar la información recolectada en el presente trabajo son las siguientes:

- Experimentación en el entorno de trabajo de los softwares utilizados.
- Comparación de los resultados obtenidos mediante el ingreso de información en los softwares.
- Creación de maquetas a escala para examinar y definir posibles maneras de conformar la estructura del conjunto diseñado.

### 1.5.5. ACTIVIDADES REALIZADAS

El proyecto se realizó considerando el siguiente orden de actividades:

- Se analizarán los diferentes tipos de herramientas digitales que se utilizan en la actualidad para el diseño paramétrico, se examinarán sus capacidades, ventajas y desventajas para definir una herramienta con la cual se trabajará.
- Definir una idea formal, a partir de la cual se abstraerá el concepto para la generación del diseño, para posteriormente definir las condiciones geométricas y matemáticas que determinarán el diseño.
- Crear un algoritmo que en base a las condiciones geométricas y matemáticas que se han definido, genere el diseño del espacio, para posteriormente explorar los diferentes

resultados mediante la iteración del algoritmo y diferentes valores para los parámetros del mismo.

- Realizar una maqueta a escala 1:1, la misma que servirá como base para el análisis del proyecto completo.



## CAPÍTULO II

### 2. DISEÑO DE UN ESPACIO MEDIANTE EL USO DE ALGORITMOS GENERATIVOS

#### 2.1. DEFINICIÓN Y ABSTRACCIÓN DE LA IDEA, CONCEPTUALIZACIÓN DEL DISEÑO

Todo Hecho Arquitectónico debe resultar de una conceptualización inicial, la creación de un concepto, sin duda representa la parte artística por excelencia de la planeación (Mota, 2014).

En todo óptimo resultado arquitectónico debe registrarse una lectura clara del concepto arquitectónico del cual se partió, para mostrar el logro satisfactorio de la intencionalidad del proyectista.

La abstracción en el diseño es una de las herramientas clave para lograr que nuestros objetos de estudio, es decir, lo que estemos por diseñar, tengan las características clave que nos permitan identificarlos, reconocerlos y entender sobre lo que tratan con tan sólo mirarlos unos instantes (Flores, 2013).

##### 2.1.1. DEFINICIÓN Y ABSTRACCIÓN DE LA IDEA

###### 2.1.1.A. ESTRATEGIAS DE DISEÑO

Teniendo bien claro lo que se quiere realizar, entonces se puede llevar a cabo un proceso que conduzca hacia

la exploración de posibles soluciones del problema planteado en este proyecto.

Esto va de la mano con las aproximaciones hacia la abstracción, aquí realizamos una definición formal de nuestro objeto-estudio.

¿Cómo será?, ¿Qué forma tendrá? ¿Qué tamaño?, ¿Qué color(es)?, ¿Cuántos metros cuadrados tendrá?, ¿Cuántos pisos?, ¿Cuánta área tiene?, ¿Para qué se usará?, entre otras interrogantes.

Con el software escogido para la realización del trabajo, en este caso específico Autodesk Dynamo en conjunto con Autodesk Revit, se comienzan a definir los parámetros y las relaciones que generarán y regirán el modelo y su diseño según las características planteadas con anterioridad.

Posteriormente se debe iniciar una etapa de observación en la cual se exploran los diferentes resultados que se generan mediante la modificación de los parámetros que se han establecido, una vez analizadas los resultados de las diferentes iteraciones del algoritmo se puede hacer una comparación para definir la propuesta que sea más factible y se apegue más a los parámetros que se han definido al inicio del proyecto.

Como último paso se ha de analizar la posibilidad de ejecución en escala real

del proyecto mediante la selección de materiales adecuados para su construcción y métodos de producción en masa para las piezas que conformen el diseño, definiendo así los más adecuados según su facilidad de manejo y factibilidad económica y de ensamblaje.

El diseño se encontrará delimitado y definido por el volumen al cual se tenga que adaptar, es posible que el resultado final no tenga nada que ver con el volumen original, aparte de ser el contenedor para el diseño, esto en conjunto con las diferentes propiedades de los materiales.

Cuando se diseña, se debe definir claramente la calidad y el tipo de espacio que se pretende manejar, considerando la función, la forma, la escala, las circulaciones, la relación interior-exterior, su acción sobre los individuos, las actividades que ejecutarán los usuarios, los materiales a emplear, el estilo, el mobiliario, entre otras características.

“Es muy difícil establecer la función de la arquitectura, y agreguemos el diseño interior en términos claros y realistas con respecto a la satisfacción de las necesidades biológicas del hombre, quien es un producto social y por ende, cambian con la historia las maneras de satisfacer tales necesidades, maneras que rebasan el elemental agrado biológico, no sólo al evitar en toda construcción lo lesivo y desfavorable al cuerpo y a la vida,

sino también al buscar caminos hacia lo estético" (Acha, 2001).

La Función, se refiere al rol de los espacios dentro de la estructura del sistema, Louis Kahn los subdivide en dos y permite darles un tratamiento formal, jerarquizado y expresivo (Browlee, De Long, & Kahn, 1998).

El espacio debe ser un espacio reducido, en el cual el usuario se pueda sentir rodeado del diseño, y de todas las diferentes interpretaciones que resulten del juego de la geometría con la luz.

El espacio que sea resultado de este proyecto, debe ser capaz de transmitir diferentes sensaciones al usuario, su configuración debe ser representativa de algo a lo cual se pueda referenciar, y que pueda ser interpretado por cada persona a su propia manera.

Es por esto que se propone la realización de una especie de pabellón o pasaje por el cual el usuario pueda recorrer e interpretar el movimiento de la superficie del diseño a su propia manera, y que esto pueda cambiar conforme el punto de vista del observador cambia en el espacio.

### 2.1.1.B. IDEA GENERAL

El trabajo se ha desarrollado a partir de la idea básica del agua, y a partir de este principio se ha desarrollado y abstraído el concepto para la generación del diseño.

El agua es una sustancia que se encuentra formada por dos átomos de hidrógeno y uno de oxígeno,  $H_2O$ , generalmente se refiere de esta manera a la sustancia en estado líquido, ya que posee los tres estados de la materia.



**Imagen 2.1.** Propagación de ondas en la superficie del agua luego de haber recibido un estímulo - Fotografía (Averill & Eldredge, 2016)

La idea principal se basó en el agua y su movimiento, debido a las propiedades específicas del agua como fluido, al ser este expuesto a un estímulo se generan ondas en su superficie, estas ondas se propagan de tal manera que afectan más la superficie del fluido cerca al estímulo y se van atenuando conforme se alejan del mismo, las ondas y su propagación pueden ser definidas matemáticamente gracias a la siguiente ecuación:

$$\eta(x,t) = a \sin(\theta(x,t))$$

De donde:

$$\theta = 2\pi(x/\lambda - t/T) = kx - \omega t$$

$$k = 2\pi/\lambda$$

$$\omega = 2\pi/T$$

- $\eta$  es la elevación de la superficie del agua (en metros).
- $a$  es la amplitud (en metros).
- $\theta$  es la función de fase (en radianes).
- $x$  es la posición horizontal (en metros).
- $t$  es la posición horizontal (en segundos).
- $\lambda$  es la longitud de onda (en metros).



- $T$  es el periodo (en segundos).
- $k$  es el número de olas (en radianes por metro).
- $\omega$  es la frecuencia angular (en radianes por segundo).

La idea principal es aplicar esta especie de movimiento que se genera en la superficie del agua en la superficie del espacio que se pretende diseñar en este trabajo. Es así que el aspecto general y volumetría del diseño estaría íntimamente relacionado y vinculado al agua en movimiento.

Siendo así se ha tomado el concepto de la estructura molecular del agua, para aplicarla de igual manera dentro del diseño, como módulo básico estructural del espacio diseñado.

Como se dijo anteriormente las moléculas de agua están compuestas por dos

átomos de hidrógeno y uno de oxígeno, y la manera en las que estas se organizan para formar la molécula de agua define un tetraedro casi regular. Conteniendo al átomo de oxígeno en el centro de masas o centro de gravedad del tetraedro formado, mientras que en los vértices se encuentran los dos átomos de hidrógeno y dos pares de electrones solitarios del átomo de oxígeno.

Más interesante aún, es la manera en la cual se unen las moléculas de agua entre sí, puesto que para unirse se agrupan de la misma manera que una molécula de agua, se asemeja a un tetraedro, en donde, en el centro de masas se encuentra una molécula de agua, que se vincula con otras moléculas de agua, que se encuentran en los vértices del tetraedro, mediante lo que se conoce como enlace de hidrógeno.

Luego de un proceso de abstracción de estos criterios, se aplicarán al diseño.

### 2.1.1.c. ABSTRACCIÓN DE LA IDEA

Los conceptos explicados en el punto anterior tienen la necesidad de pasar por un proceso de abstracción para poder ser aplicados al diseño, el primer paso para esto, es definir ciertas interrogantes sobre el diseño del espacio que será resultado del presente trabajo.

Tras haber hecho una investigación extensa sobre la aplicación del diseño paramétrico y su aplicación en la arquitectura, se ha establecido que lo más apropiado y manejable para el trabajo de investigación es manejar una escala pequeña, esto se refiere a espacios habitables pequeños, que puedan albergar diferentes actividades en su interior, sin que se piense específicamente en una. El tema del tamaño se vuelve un factor muy importante a considerar, sobre todo por el hecho de facilitar la utilización de diferentes materiales livianos.

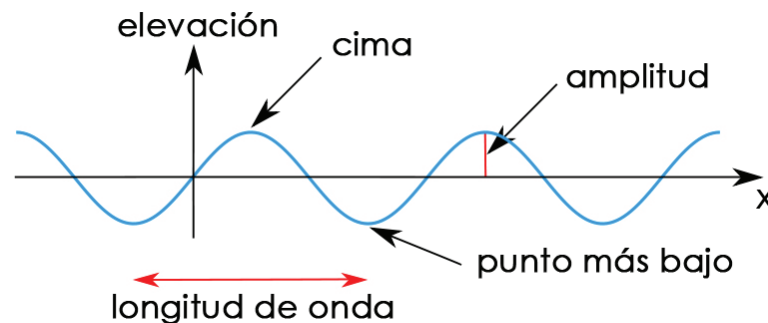


Imagen 2.2. Representación de la propagación de ondas en la superficie del agua - Imagen digital (Wikipedia, 2016)

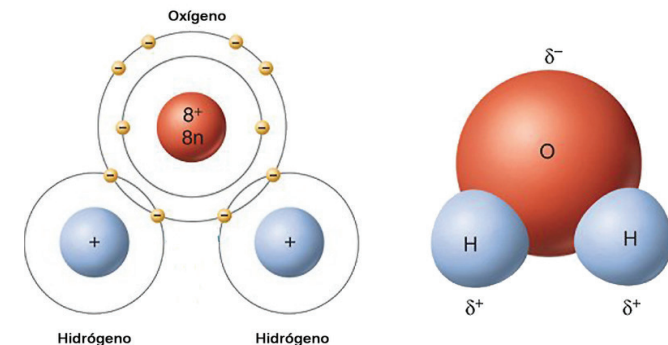
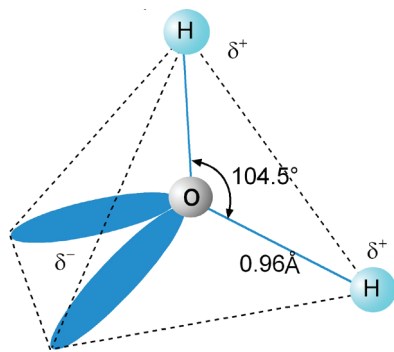
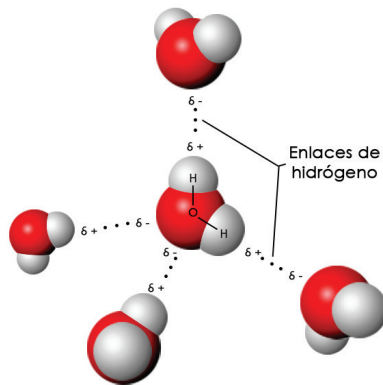


Imagen 2.3. Molécula de agua, diferentes representaciones - Imagen digital (Bonsai Menorca, 2015)

Es así que se ha definido que el área en planta no deberá ser mayor a 24 m<sup>2</sup>. Con el fin de permitir la utilización de materiales livianos, como por ejemplo el cartón prensado, se ha establecido que el espacio será interior, es decir que no se lo colocará en exteriores o a la intemperie, permitiendo además de esta manera que



**Imagen 2.4.** Estructura tetraédrica de la molécula de agua - Imagen digital (Ciencia y Salud, 2012)



**Imagen 2.5.** Enlaces entre moléculas de agua, enlaces de hidrógeno - Imagen digital (QuimiTube.com, s.f.)

el diseño resulte en una especie de "piel" o recubrimiento que se pueda aplicar a espacios interiores ya definidos.

Es así que partiendo de la premisa que el diseño se puede aplicar a espacios interiores ya construidos y definidos, el primer paso será poder ingresar la información de estos espacios al algoritmo que se creará, y mediante la aplicación de modificadores, transformar el espacio en uno diferente, que presente las características del movimiento de la superficie del agua. Otro punto importante, es la posibilidad de la aplicación del algoritmo a espacios no ortogonales.

El algoritmo que se genere deberá entonces de alguna manera trasladar el movimiento de las ondas sobre la superficie del agua a todas las caras que delimiten el espacio en el cual se insertará el diseño. Siendo este el todo, ahora se deben definir los elementos que comprenden el todo, a manera de un módulo básico que se adapte al movimiento de estas superficies.

Se plantea definir el espacio en donde se emplazará el diseño, para posteriormente dividirlo en secciones que se desarrollarán a lo largo de un eje en el espacio. Estas secciones serán luego reemplazadas con la ecuación de la curva generada por las ondas en la superficie del agua, las curvas se irán perdiendo poco a poco conforme se acerque a la última sección, dando así la sensación de movimiento a la superficie.

La superficie que resulte del proceso anterior se debe abstraer para poderse construir. La idea para generar los elementos que conformen la superficie es similar a la de la composición molecular del agua, las moléculas de agua tienen una forma tetraédrica, que es un sólido con cuatro caras triangulares.

Siendo el triángulo la figura geométrica más estable estructuralmente, pues no se puede deformar, se ha decidido generar una triangulación sobre la superficie definida. Como se vio en el capítulo anterior una de las triangulaciones más útiles es la de Delaunay, por lo que este será el criterio que se aplicará al diseño.

Una vez definida la triangulación sobre la superficie, cada uno de los triángulos creados pueden ser utilizados como base para la colocación del módulo básico que tiene su origen en la abstracción del tetraedro.

El módulo básico se deriva de la organización molecular del agua, es decir de un tetraedro, la base del módulo es triangular, siendo los vértices de esta base los que se conectarán con los vértices de otros módulos para crear una superficie. Tomando el triángulo que define la base del módulo se toma la altura a cada uno de los lados del triángulo, se saca el promedio, y esta medida se utiliza como altura para crear un tetraedro, el punto de inicio es el baricentro del triángulo base.

## 2.2. ESTABLECIMIENTO DE CONDICIONES GEOMÉTRICAS Y MATEMÁTICAS

Habiendo decidido ya el concepto que se utilizará para el desarrollo del diseño, se hace evidente la necesidad de establecer las condiciones tanto geométricas como matemáticas que regirán al mismo, para posteriormente poder trasladar estas condiciones y reglas al programa en el que se desarrollará el algoritmo, es decir Dynamo.

De igual manera que los algoritmos presentan un conjunto de instrucciones ordenadas paso a paso para llegar a un fin específico, se establecerán las condiciones de manera ordenada, y conforme sea necesario se crearán nuevas condiciones o relaciones para continuar con el proceso.

A continuación, se explicará de manera detallada las condicionantes del diseño.

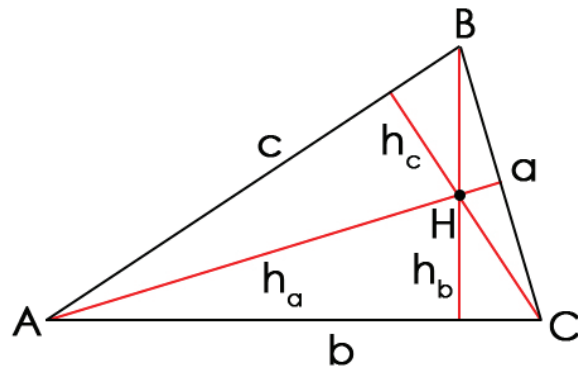


Imagen 2.6. Alturas de un triángulo y su ortocentro, punto H - Imagen digital (Elaboración propia, 2016)

### 2.2.1. ESPACIO CONTENEDOR DEL DISEÑO

Lo primero que se necesita definir es el espacio o ambiente en el cual se emplazará el diseño. Se ha establecido que el ambiente tendrá como base un cuadrilátero, es decir cuatro lados, el mismo que puede ser regular o no, luego se toman los vértices del cuadrilátero y se duplican y se les agrega un modificador para definir la altura de cada uno, de esta manera estos cuatro nuevos vértices crean la cubierta del espacio.

El cuadrilátero ABCD define la base del volumen, mientras que el cuadrilátero A'B'C'D' define la cubierta del mismo. Se establece entonces que el punto A se encuentra en el origen, coordenadas  $x=0, y=0, z=0$ , a partir de este punto se determinará la ubicación de los puntos B, C y D mediante el ingreso de información directamente al programa.

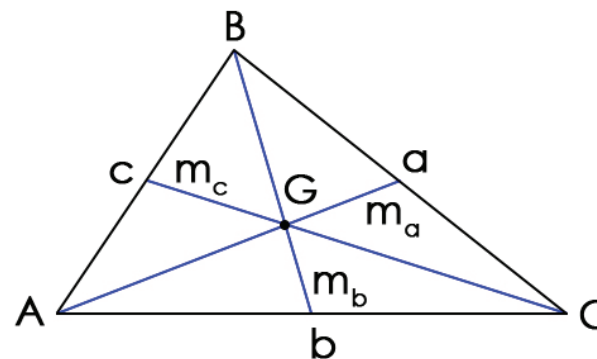


Imagen 2.7. Medianas de un triángulo y su baricentro, punto G - Imagen digital (Elaboración propia, 2016)

Los puntos A', B', C' y D' se colocarán exactamente en los puntos A, B, C y D respectivamente, y se les establecerá un nuevo sistema de coordenadas a cada uno con base en los puntos A, B, C y D, de esta manera cualquier modificación en alguno de los puntos de la cubierta estará directamente relacionada con sus contrapartes de la base. Es decir, para el punto B', su origen estará ubicado en el punto B, mientras que el punto B tiene su origen en el punto A.

Se puede entonces definir la ubicación de cada uno de los puntos en la cubierta mediante su desplazamiento en x, y, z, en relación a los puntos de la base.

Estas condiciones permiten la creación de volúmenes con caras inclinadas y no ortogonales.

### 2.2.2. SECCIONES DEL ESPACIO CONTENEDOR

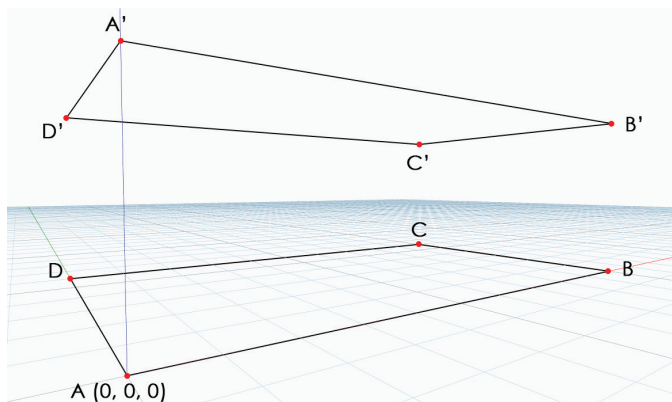
Tras haber definido y delimitado el espacio que contendrá al diseño, se procede a dividir al mismo para generar secciones.

Las secciones se crean solamente en una de las direcciones, se ha establecido que estas se desarrollarán a lo largo de las líneas conformadas entre los puntos AB, CD, A'B' y C'D'. Se procede a dividir en partes iguales los lados AB, CD, A'B' y C'D', con el fin de que se pueda modificar posteriormente de ser necesario, se añade

un parámetro a el número de divisiones que se desea generar.

No se consideran ni el punto de inicio ni el final de los lados para la creación de las secciones, solamente el tramo entre estos puntos, por lo que se establece como regla que, si se desea generar  $n$  número de secciones, es necesario dividir la línea, AB, por ejemplo, en  $n+1$  partes iguales, el resultado es  $n$  puntos intermedios entre el punto de inicio y el final.

Al dividir las líneas AB, CD, A'B' y C'D', se establecen  $n$  número de puntos en cada una de las líneas, al unir estos puntos con su contraparte de la línea adyacente, el primer punto creado entre AB y el primer punto creado entre A'B' por ejemplo, y así consecutivamente hasta regresar a la primera línea, el resultado será finalmente  $n$  número de secciones deseadas.



**Imagen 2.8.** Base y cubierta del espacio contenedor del diseño - Captura de pantalla (Elaboración propia, 2016)

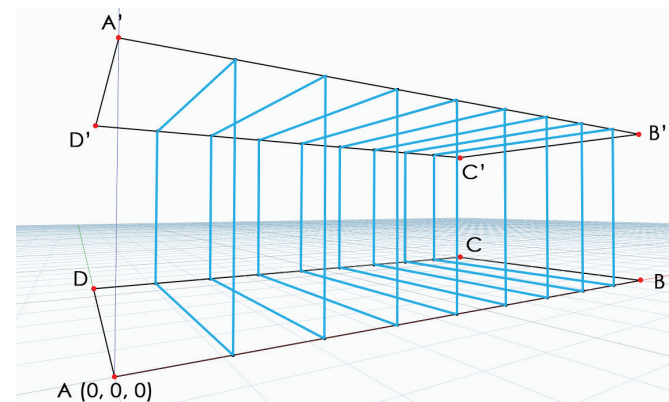
### 2.2.3. MODIFICACIÓN DE LAS SECCIONES DEFINIDAS

El próximo paso es modificar las secciones que se consiguieron en anteriormente. Primero que nada, cabe recalcar que hasta este punto las secciones están constituidas por varias líneas, mientras que para poder utilizarlas para generar una superficie deben ser un solo elemento, siendo así se ha establecido que las secciones deben ser representadas por curvas, el tipo de curva más apropiado para el efecto que se desea conseguir es el tipo NURBS. Las curvas NURBS trabajan con lo que se conoce como puntos de control, al ser el resultado de una interpolación la curva en sí no pasa necesariamente sobre los puntos de control.

Puesto que el diseño se piensa aplicar solamente en las caras laterales y cubierta

del espacio, solamente tomaremos las líneas que se encuentren en estas caras, en este caso particular son las comprendidas entre AB-A'B', A'B'-C'D'C y C'D'-CD. Sobre de estas líneas se desea colocar puntos para utilizarlos como puntos de control para la generación de la curva, se aplica entonces el mismo procedimiento que para las secciones, cada una de las líneas a intervenir se divide en partes iguales, se define  $n$  número de puntos a colocarse entre el punto inicial y final, mientras que la longitud de cada línea se dividirá para  $n+1$  número de secciones iguales.

El resultado es un conjunto de puntos que contiene  $n(AB-A'B') + n(A'B'-C'D') + n(C'D'-CD)$  número de puntos, además se debe incorporar los puntos en AB y CD de cada sección, es decir que al número anterior se debe añadir dos puntos más.



**Imagen 2.9.** Secciones creadas en el espacio contenedor del diseño - Captura de pantalla (Elaboración propia, 2016)



Cada sección tiene ahora un conjunto de puntos ordenados, a partir de estos se crea una curva NURBS para cada una de las secciones,

Cada sección tiene ahora un conjunto de puntos ordenados, ahora se deben desplazar hacia el interior del espacio, de la misma manera que en los pasos anteriores, se crea un parámetro que pueda ser modificado con facilidad, la longitud que se desplaza cada punto,  $d$ , determina la distancia a la cual el punto se encuentra de la cara del espacio desde el cual partió.

Con los nuevos conjuntos de puntos se crea una curva NURBS para cada una de las secciones, desde este paso se podría crear ya una superficie a partir de las secciones que se han definido, sin embargo, falta incorporar el movimiento de las ondas.

### 2.2.3.A. DEFINICIÓN DE LA ECUACIÓN DE LA ONDA

La ecuación que se utilizó es la misma definida anteriormente, reemplazando variables y simplificando la ecuación es la siguiente:

$$\eta = \alpha \sin(kx - \omega t)$$

Recordemos que:

$$k = 2\pi/\lambda$$

$$\omega = 2\pi/T$$

La variable  $\eta$  es la que define cuanto se desplaza el punto en el eje  $y$  y a partir de  $y=0$ , esta variable se encuentra en función de otras variables, las mismas que se ingresarán manualmente para el cálculo de  $\eta$ .

Se definen las variables  $\alpha$  (amplitud),  $x$  (posición horizontal en función del espacio),  $t$  (posición horizontal en función del tiempo),  $\lambda$  (longitud de onda),  $T$  (periodo),  $k$  (número de olas),  $\omega$  (frecuencia angular), estas variables son las encargadas de definir la curva, por lo que se debe crear un parámetro para cada una.

Se debe señalar también que la ecuación descrita anteriormente se aplica para ondas constantes que no presentan variación en sus elementos, por lo que se ve la necesidad de establecer dos parámetros para cada variable, uno para su valor inicial y otro para su valor final, de esta manera la curva generada será cambiante en su desarrollo en relación a los valores ingresados.

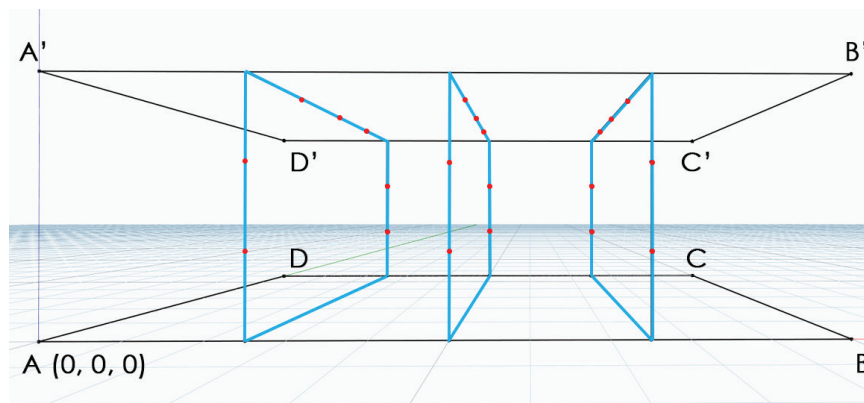


Imagen 2.10. División de las secciones y creación de puntos de control - Captura de pantalla (Elaboración propia, 2016)

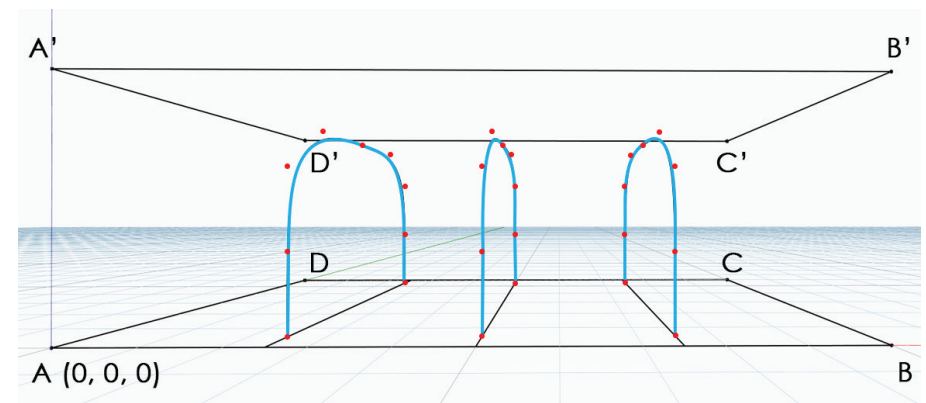


Imagen 2.11. Desplazamiento de los puntos de control, definición de curvas NURBS - Captura de pantalla (Elaboración propia, 2016)

### 2.2.3.B. TRASLACIÓN DE LAS ONDAS A LAS SECCIONES

Para poder incorporar los valores obtenidos con la ecuación de las ondas, primero debemos dividir las secciones definidas por las curvas NURBS en partes iguales, para crear  $n$  número de puntos al largo de su recorrido, asignando a cada punto un valor para  $x$ , estos puntos posteriormente se desplazarán hacia el interior o exterior del espacio, dependiendo del valor de  $\eta$  en  $x$ . Si el valor de  $\eta$  es positivo, se desplazará hacia el interior del espacio, si es negativo, se desplaza hacia el exterior.

El resultado del desplazamiento de los puntos es un nuevo conjunto de puntos, que en su recorrido refleja el movimiento de las ondas según los parámetros definidos en la ecuación. Mientras más puntos se creen la interpolación dará como resultado una

curva más suave y definida, mientras que con menos puntos puede hasta perder la forma deseada.

Finalmente se debe crear una curva a través de los puntos definidos, de igual manera que en el punto anterior se utilizarán curvas NURBS. Quedan así definidas las secciones para generar la superficie.

### 2.2.4. CREACIÓN DE LA SUPERFICIE DEL DISEÑO

Luego de haber definido las secciones definitivas, se debe determinar las condiciones para generar la superficie, en este punto hay tres posibilidades para generar la superficie.

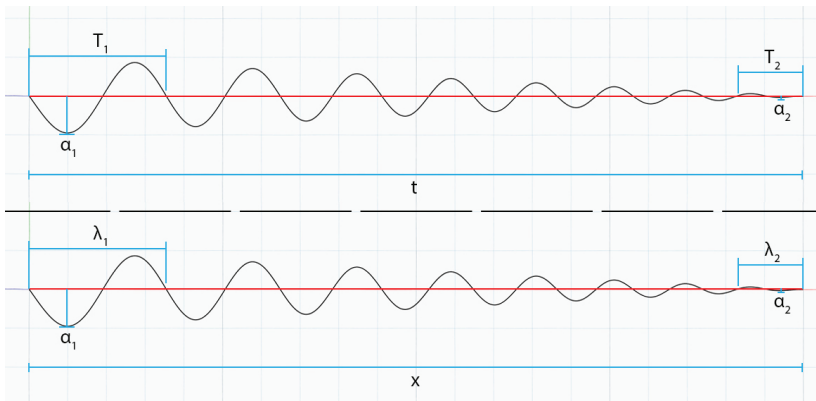
Se puede interpolar todas las secciones creadas sin considerar los lados BC y DA, el resultado de esto será una superficie que

se asemeja a un túnel, pues se encuentra abierta por los dos extremos.

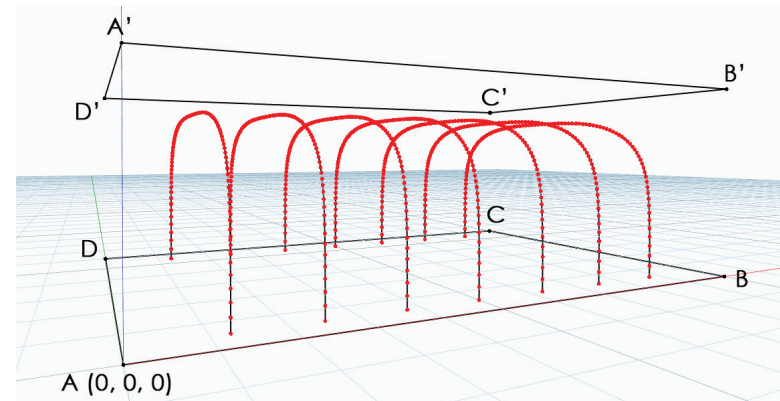
Otra opción es interpolar las secciones definidas, pero se considera uno de los lados dentro de los elementos a interpolar, ya sea el BC o DA, el resultado será similar al primero, pero uno de los extremos se encuentra cerrado.

La última opción es considerar tanto el lado BC como el DA para la interpolación de las secciones, de esta manera el resultado será un espacio completamente cerrado de todas las caras.

Es evidente que el algoritmo debe ser capaz de representar cualquiera de estas tres opciones, por lo que se debe incorporar un parámetro para determinar cuál de las tres posibilidades se realizará.



**Imagen 2.12.** Gráficos de la ecuación de la onda, en función del tiempo  $t$  y el espacio  $x$  - Captura de pantalla (Elaboración propia, 2016)



**Imagen 2.13.** Creación de  $n$  número de puntos a lo largo de las secciones NURBS - Captura de pantalla (Elaboración propia, 2016)



### 2.2.5. TRIANGULACIÓN DE LA SUPERFICIE

Una vez definida la superficie, se ha de aplicar una triangulación de Delaunay a la misma, para esto el primer paso es definir n número de puntos a generarse sobre la superficie, el número de puntos se establece como un parámetro, que puede ser modificado con facilidad posteriormente.

Debido a la irregularidad de la superficie definida, la forma más fácil de definir puntos que pertenezcan a la misma, es mediante la utilización de la representación de un dominio utilizando los parámetros U, V, cada uno de los puntos definidos tendrá sus propios valores para U y V.

Dada la gran cantidad de puntos que se pueden llegar a generar, en los rangos de cientos a miles dependiendo de la cantidad

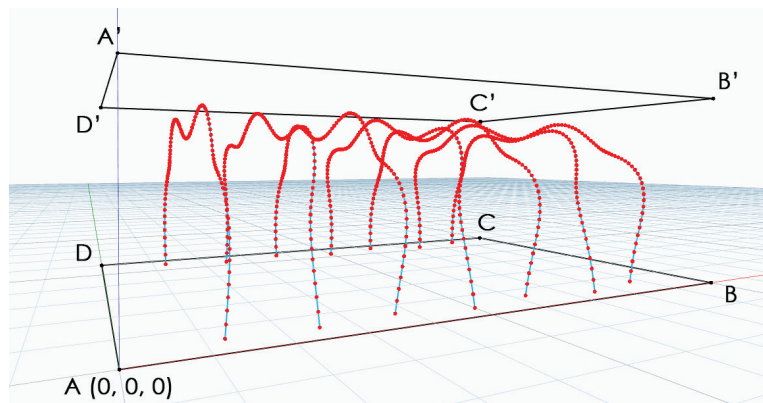
de triángulos que se deseen crear, no es factible ingresar manualmente los valores para U y V de cada uno de los puntos, por eso se ha de establecer un comando que permita generar valores aleatorios para cada punto. Al ser valores aleatorios, existe la posibilidad de que algunos puntos se encuentren demasiado cerca uno de otro, como para que el elemento que se genere sea construible o pueda siquiera ser considerado sin generar algún tipo de error, por lo que es necesario agregar un parámetro que permita definir un rango mínimo al cual deben estar un punto de cualquier otro dentro del conjunto.

Una vez definido el conjunto de puntos, se ejecuta el algoritmo para crear la triangulación, el mismo que viene incorporado dentro del software, el resultado serán un conjunto de triángulos.

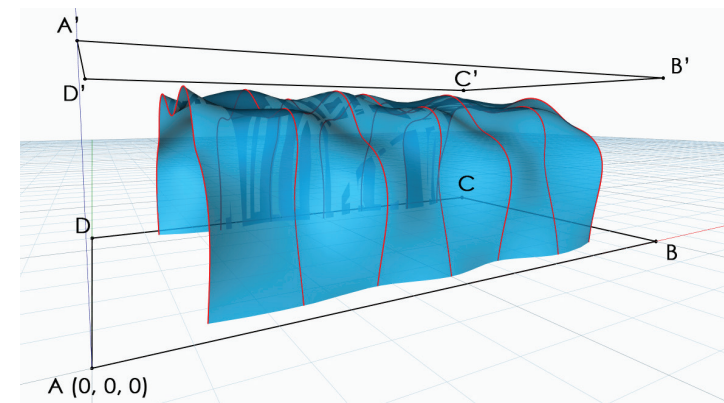
### 2.2.6. DEFINICIÓN DE LOS PANELES

Los triángulos definidos en el punto anterior que son resultado de la aplicación de una triangulación de Delaunay, pueden o no ser regulares, aunque lo más seguro y probable es que no existan triángulos regulares dentro del conjunto, se debe entonces buscar un modelo matemático para los paneles que se pueda aplicar a cualquier triángulo.

El módulo básico está basado en un tetraedro, la base del mismo definida por el triángulo abc, el baricentro del triángulo es el punto G, definido por la intersección de las medianas, la línea que une un vértice con el punto medio del lado opuesto al vértice,  $am_a$ ,  $bm_b$  y  $cm_c$ . Para definir el punto d, vértice superior del tetraedro, se realiza un promedio de las tres alturas del triángulo base, la altura es la línea que une



**Imagen 2.14.** Secciones definitivas, mediante el uso de curvas NURBS - Captura de pantalla (Elaboración propia, 2016)



**Imagen 2.15.** Superficie creada mediante la interpolación de las curvas en las secciones - Captura de pantalla (Elaboración propia, 2016)

un vértice y el lado opuesto del mismo de manera perpendicular, el valor resultado de esta operación es la distancia a la cual el punto d se encuentra del triángulo base, de esta manera no solo cada triángulo dentro de la superficie será único, sino que el tetraedro será directamente definido por las dimensiones del triángulo.

El siguiente paso es definir el centro de masas del tetraedro, punto M, este punto se consigue al trazar líneas desde cada uno de los vértices del tetraedro hacia los baricentros de las caras opuestas a los vértices, la intersección de todas estas rectas define el centro de masas.

Finalmente, los elementos físicos del módulo están definidos por los triángulos aMb, bMc y cMa, estos triángulos tendrán como puntos de inserción los vértices definidos en la triangulación de la superficie.

### 2.2.7. COLOCACIÓN DE LOS PANELES

Una vez definidos los paneles, estos se pueden colocar ya en el lugar de los triángulos que conforman la superficie, para lo cual se debe definir una lista en la cual se agrupen los vértices de cada triángulo como un subconjunto, para poder colocar sobre estos el panel.

A los paneles se les agrega un parámetro adicional, este parámetro recogerá valores del 0 al 1, estos valores estarán dados por la relación entre el vector normal definido en el baricentro de los triángulos base y el vector definido por la incidencia solar sobre el entorno, de acuerdo al grado de perpendicularidad que tengan las caras de los triángulos con el vector de la incidencia solar, el valor será 1 si es la cara es perpendicular al vector, mientras que si es paralela el valor será 0, así se irá

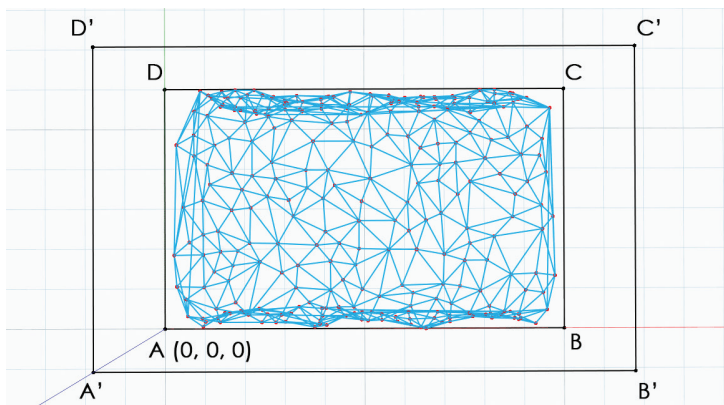
determinando valores intermedios acorde a la relación existente.

El resultado del valor del parámetro permitirá añadir a los paneles una abertura, el tamaño de la misma estará en relación directa con el valor del parámetro.

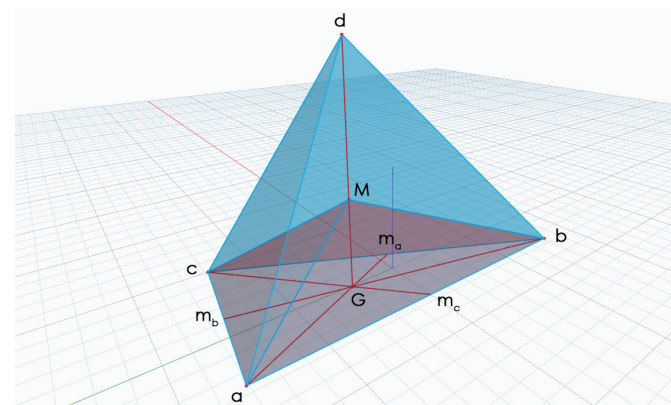
### 2.2.8. SOFTWARES UTILIZADOS PARA EL PROYECTO

#### 2.2.8.A. AUTODESK REVIT

El software Revit fue creado con la visión BIM en mente, todos los componentes que se pueden crear dentro del programa tienen la posibilidad de almacenar diferente tipo de información, la misma que puede ser útil para tanto para la puesta en obra del proyecto, la coordinación con otras disciplinas y el control de funcionamiento del diseño.



**Imagen 2.16.** Triangulación de Delaunay aplicada a la superficie del diseño - Captura de pantalla (Elaboración propia, 2016)



**Imagen 2.17.** Representación básica de los paneles a ser colocados - Captura de pantalla (Elaboración propia, 2016)

Es una aplicación que admite un flujo de trabajo BIM que va desde la etapa conceptual a la etapa de construcción. La finalidad de Revit es crear modelos con una precisión muy alta, al mismo tiempo que optimiza el rendimiento y facilita la colaboración entre los diferentes equipos de trabajo en un proyecto.

Revit permite un desarrollo de manera completa el proyecto arquitectónico a partir de un modelo tridimensional, necesitando poca interacción por parte del usuario para generar la documentación relativa al proyecto constructivo. Optimiza el flujo de trabajo al recopilar información detallada acerca de los elementos constructivos y los cálculos de cantidades.

La posibilidad de realizar cambios en cualquier etapa del proyecto, sin que esto afecte la productividad o llegue a incidir

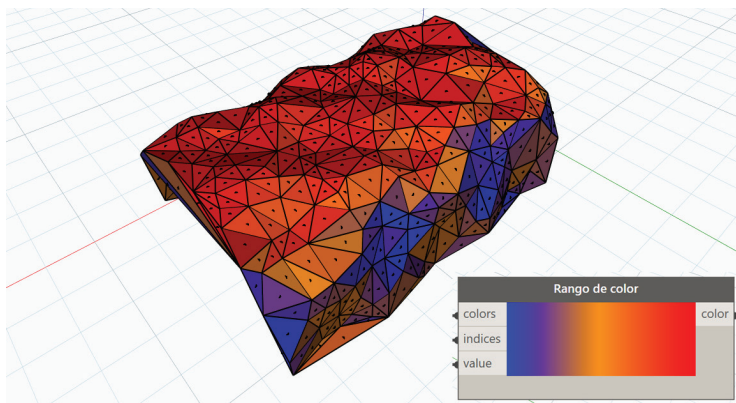
en los costos o tiempos de entrega, es una de las grandes ventajas de los softwares BIM sobre los CAD y específicamente de Revit. Permite compatibilidad total entre las diferentes disciplinas e interacción con todos los profesionales que intervienen en la creación del proyecto desde una misma plataforma.

### 2.2.8.B. AUTODESK DYNAMO

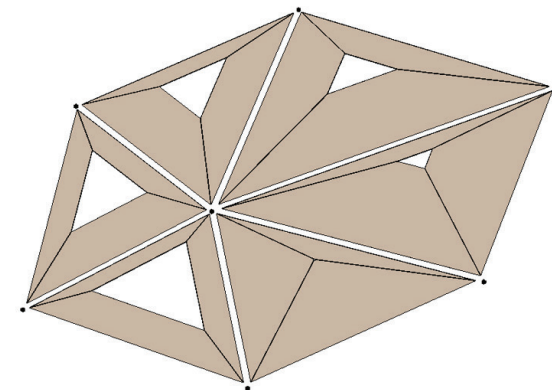
Dynamo es un plug-in de diseño generativo para Revit, que hace posible la creación de nuevas posibilidades en el diseño paramétrico. Es un software de diseño computacional algorítmico en 3D, todo aquello que modela posee propiedades paramétricas. Posee aplicaciones de código abierto, lo cual permite la creación de herramientas personalizadas por parte de los usuarios (Huang, 2014).

En las últimas entregas del software este puede descargarse y ejecutarse como un programa aparte en lugar de ser un plug-in, el nombre de este paquete de software es Autodesk Dynamo Studio. Es una herramienta de programación visual que pretende ser accesible tanto para los no programadores y programadores por igual. Se ofrece a los usuarios la capacidad de definir visualmente el comportamiento de los elementos modelados, definir piezas a la medida de la lógica y secuencia de comandos con varios lenguajes de programación textuales (GitBook, 2015).

Una vez que hemos instalado la aplicación, Dynamo nos permitirá trabajar dentro de un entorno de programación visual en el que conectaremos elementos para definir las relaciones y las secuencias de acciones que componen algoritmos personalizados. Se pueden utilizar estos algoritmos para



**Imagen 2.18.** Codificado por color según la perpendicularidad de la cara en relación al vector definido por la incidencia solar - Captura de pantalla (Elaboración propia, 2016)



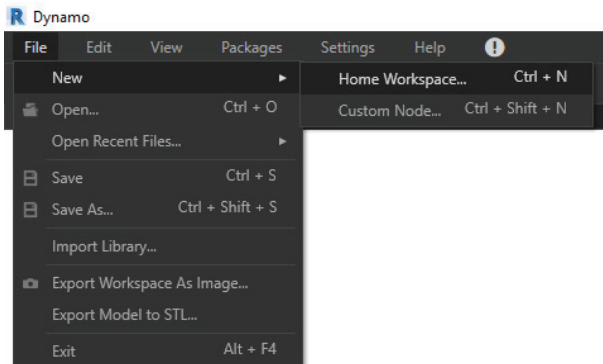
**Imagen 2.19.** Abertura generada en los paneles, valores 0, 0.2, 0.4, 0.6, 0.8, 1 - Captura de pantalla (Elaboración propia, 2016)



una amplia gama de aplicaciones de procesamiento de datos o para la generación de geometría, todo en tiempo real y sin necesidad de escribir códigos.

Mientras Dynamo es un entorno flexible, diseñado para funcionar con una amplia gama de programas, se creó originalmente para su uso con Revit, es así que Dynamo posee todo un conjunto de nodos diseñados específicamente para Revit.

Mediante el uso de los nodos de Dynamo en conjunto con los elementos parametrizables de Revit, un usuario puede mejorar considerablemente los flujos de trabajo, parámetros relativos a la interoperabilidad, documentación, análisis y generación pueden incorporarse en algoritmos creados en Dynamo. Ciertos flujos de trabajo se pueden automatizar, mientras que la exploración opciones de diseño pueden facilitarse.



**Imagen 2.20.** Menú para crear un nuevo proyecto - Captura de pantalla (Elaboración propia, 2016)

### 2.3. DEFINICIÓN DEL ALGORITMO GENERATIVO PARA EL DISEÑO

Una vez definidas todas las condiciones matemáticas y geométricas que regirán al diseño, es necesario ingresarlas dentro del software Dynamo para poder generar un conjunto de instrucciones, el algoritmo, que generará un diseño a partir de los parámetros ingresados, al modificarse los parámetros el diseño cambiará para representar la nueva geometría definida por la nueva iteración.

Dynamo permite agrupar los nodos para un manejo más cómodo y fluido del algoritmo, es así que de acuerdo a la función que cumple cada conjunto de nodos, se han establecido cuatro tipos principales para el desarrollo del presente trabajo.

#### Grupo de nodos modificadores

Estos nodos están dedicados específicamente al ingreso directo de información para el programa, el usuario final podrá especificar el valor para cada parámetro en estos nodos, ya sean valores numéricos o booleanos, en el proyecto se ha definido un color azul claro para diferenciar de otros grupos.

#### Grupo de nodos de procesos

Los nodos de procesos se enfocan en el procesamiento de la información ingresada en los modificadores, no requieren de

ninguna interacción con el usuario pues todas las conexiones y definiciones ya se han establecido, se les ha asignado un color verde claro para su representación.

#### Grupo de nodos de resultados

Se han creado específicamente para agrupar los resultados de los grupos de procesos, generalmente se utiliza el nodo "Watch" para extraer el resultado de una operación y colocar este resultado cerca de un grupo de procesos que requiera esta información, se ha establecido un color naranja claro para distinguirlo.

#### Grupo de nodos condicionales

Estos grupos poseen nodos que procesan información de tipo booleana, este tipo de información se debe manejar con mucho cuidado, ya que un error puede generar un fallo crítico en el software, por lo que se le ha asignado un grupo específico para facilitar el manejo y modificación de sus procesos, se ha asignado un color rosado claro para diferenciarlos.

#### 2.3.1. INSTRUCCIONES PARA DYNAMO

Lo primero que se hace dentro del entorno de trabajo de Dynamo es crear una definición sobre la cual colocaremos los nodos para establecer las instrucciones que el programa debe seguir para generar el diseño.

Dentro del programa, en la pestaña File, se hace clic en la opción New y finalmente en el recuadro que dice Home Workspace, alternativamente se puede presionar al mismo tiempo las teclas Ctrl + N, se nos abrirá el entorno de edición gráfico de Dynamo, un espacio en blanco y un plano cartesiano con los ejes x (línea en rojo), y (línea en verde), z (línea en azul).

Dynamo presenta dos tipos de pantallas sobre las que se puede trabajar, la primera y que se encuentra activa por defecto al crear un nuevo proyecto, es la Vista de Nodos (Node View), en esta vista se colocan todos los nodos y se realizan las conexiones entre ellos mediante los cables, y la segunda es la Vista de Geometría (Geometry View), en esta vista Dynamo representa todo lo que es resultado de

la ejecución del código definido por los nodos, no se puede editar directamente, es más una manera de visualizar la geometría que se ha creado. Para pasar de una vista a la otra se puede hacer clic derecho en el espacio de trabajo y alternativamente hacer clic en Switch to Geometry View o Switch to Node View, dependiendo de la vista que se encuentre activa en ese momento, también se puede presionar Ctrl + B para directamente alternar entre las vistas.

Se ha establecido una nomenclatura específica para poder denotar a los nodos, primero se ingresa la descripción del nodo seguido del tipo de nodo entre paréntesis, por ejemplo, "Nodo de prueba (Categoría. Nombre)".

### 2.3.1.A. DEFINICIÓN DE LA BASE DEL VOLUMEN

Se requiere definir los cuatro puntos que conforman el cuadrilátero de la base, A, B, C y D.

Primero se define el punto A, para esto se utiliza un crea el nodo "A (Point.ByCoordinates)", este nodo nos permite definir un punto en el espacio a partir de sus coordenadas x, y, z. El punto A se debe encontrar en las coordenadas (0, 0, 0), en Dynamo si a una entrada numérica no se le asigna un valor, generalmente por defecto el valor es 0, por lo que no se les ingresan valores a las entradas del nodo, la salida de este nodo es el punto A de coordenadas (0, 0, 0).

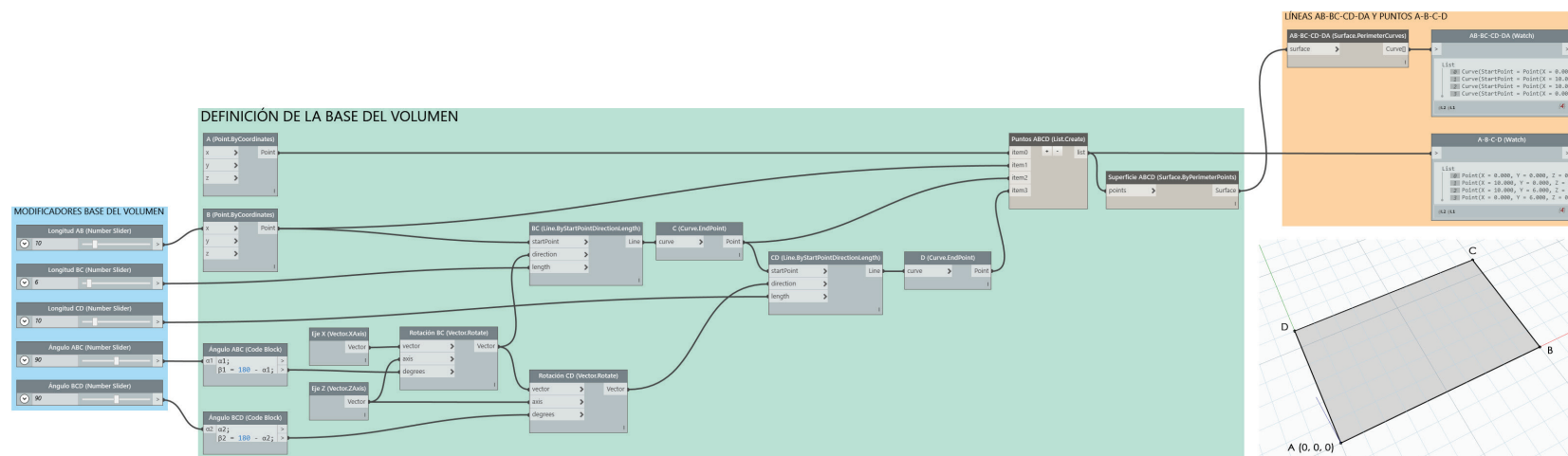


Imagen 2.21. Algoritmo que define la base del volumen y sus modificadores - Captura de pantalla (Elaboración propia, 2016)



Se crea el nodo "B (Point.ByCoordinates)" para definir el punto B, el punto en sí debe encontrarse sobre el eje x, por lo cual la distancia a la cual se encuentra de A es igual a la coordenada x que se ingrese en la respectiva entrada del nodo, es así que el punto queda definido por las coordenadas B (x, 0, 0). Se conecta el nodo "Longitud AB (Number Slider)" a la entrada que define la coordenada x, este nodo permite utilizar un control que se desliza de izquierda a derecha para definir valores numéricos, los mismos que están establecidos por un rango comprendido entre un valor mínimo y un valor máximo, además de permitir definir el monto en el que se incrementa o disminuye el valor con cada movimiento, en este caso específico se establece un rango entre los valores de 1 a 50 y con incrementos de 1.

Luego se crea el nodo "Longitud BC (Number Slider)", este define la distancia entre B y C, y se lo conecta a la entrada de longitud del nodo "BC (Line.ByStartPointDirectionLength)", este último nodo permite definir una línea a partir de un punto inicial, la dirección y la longitud de la línea, en la entrada de punto de inicio se conecta el nodo "B (Point.ByCoordinates)", la entrada dirección acepta solamente información de tipo vectorial, por lo que es necesario tomar el vector direccional del eje x, en donde se encuentran A y B, para esto se crea el nodo "Eje X (Vector.XAxis)", este tipo de nodo produce como resultado un vector normalizado al eje x.

Antes de poder conectar este nodo se necesita rotarlo con respecto al eje z la cantidad deseada, para esto se crea el nodo "Rotación BC (Vector.Rotate)" y se conectan a su entrada vector el nodo "Eje X (Vector.XAxis)", a la entrada eje se conecta el nodo "Eje Z (Vector.ZAxis)" y en la entrada grados se debe ingresar el valor a rotar.

Con el fin de determinar el ángulo interior al cuadrilátero para ABC, el valor de "Ángulo ABC (Number Slider)" debe conectarse a la entrada  $\alpha_1$  del nodo "Ángulo ABC (Code Block)", la salida  $\beta_1$  de este nodo es igual a 180 menos el valor de  $\alpha_1$ , se conecta la salida  $\beta_1$  a la entrada de grados del nodo "Rotación BC (Vector.Rotate)", luego se conecta la salida vector de este nodo a la entrada dirección del nodo "BC (Line.ByStartPointDirectionLength)". El nodo "Ángulo ABC (Number Slider)" tiene un rango de 5 a 175 con incrementos de 5. La salida del nodo "BC (Line.ByStartPointDirectionLength)" se conecta con la entrada del nodo "C (Curve.EndPoint)", este tipo de nodo da como resultado el punto final de una curva.

El mismo principio se utiliza para definir el nodo "CD (Line.ByStartPointDirectionLength)". Se crea el nodo "Ángulo BCD (Number Slider)", con rango de 5 a 175 e incrementos de 5, el mismo que se conecta a la entrada  $\alpha_2$  del nodo "Ángulo BCD (Code Block)", en donde la salida  $\beta_2$  es igual a 180 menos el valor de  $\alpha_2$ ,  $\beta_2$  se conecta

a la entrada grados del nodo "Rotación CD (Vector.Rotate)", a la entrada vector de este nodo se conecta la salida vector del nodo "Rotación BC (Vector.Rotate)" y finalmente a la entrada eje se conecta la salida del nodo "Eje Z (Vector.ZAxis)". La salida vector del nodo "Rotación CD (Vector.Rotate)" se conecta con la entrada dirección del nodo "CD (Line.ByStartPointDirectionLength)", la entrada punto de inicio se conecta con la salida del nodo "C (Curve.EndPoint)", finalmente la entrada longitud se conecta con la salida del nodo "Longitud CD (Number Slider)" que tiene un rango de 5 a 175 con incrementos de 5.

Se crea el punto D mediante el nodo "D (Curve.EndPoint)", se unen todos los puntos en una sola lista al conectar sus salidas a las entradas del nodo "Puntos ABCD (List.Create)", este nodo crea una lista al unir los elementos que se conecten a sus entradas.

El siguiente paso es crear una superficie a partir de los puntos definidos, para esto conectamos la salida del nodo "Puntos ABCD (List.Create)", que contiene la lista de puntos, a la entrada del nodo "Superficie ABCD (Surface.ByPerimeterPoints)" que permite crear una superficie al ingresar los puntos que conforman el perímetro de dicha superficie.

El resultado del programa visual definido es la superficie ABCD.



Se conecta la salida del nodo "Superficie ABCD (Surface.ByPerimeterPoints)" a la entrada del nodo "AB-BC-CD-DA (Surface.PerimeterCurves)" para definir las rectas que conforman el perímetro de la superficie ABCD. Los nodos "AB-BC-CD-DA (Watch)" y "A-B-C-D (Watch)" nos permiten ver los elementos que conforman la superficie, líneas y vértices, al conectar sus entradas con las salidas de los nodos "AB-BC-CD-DA (Surface.PerimeterCurves)" y "Puntos ABCD (List.Create)" respectivamente.

Nodos modificadores en esta sección:

- Longitud AB (Number Slider)
- Longitud BC (Number Slider)
- Longitud CD (Number Slider)
- Ángulo ABC (Number Slider)
- Ángulo BCD (Number Slider)

Nodos resultados en esta sección:

- AB-BC-CD-DA (Surface.PerimeterCurves)
- AB-BC-CD-DA (Watch)
- A-B-C-D (Watch)

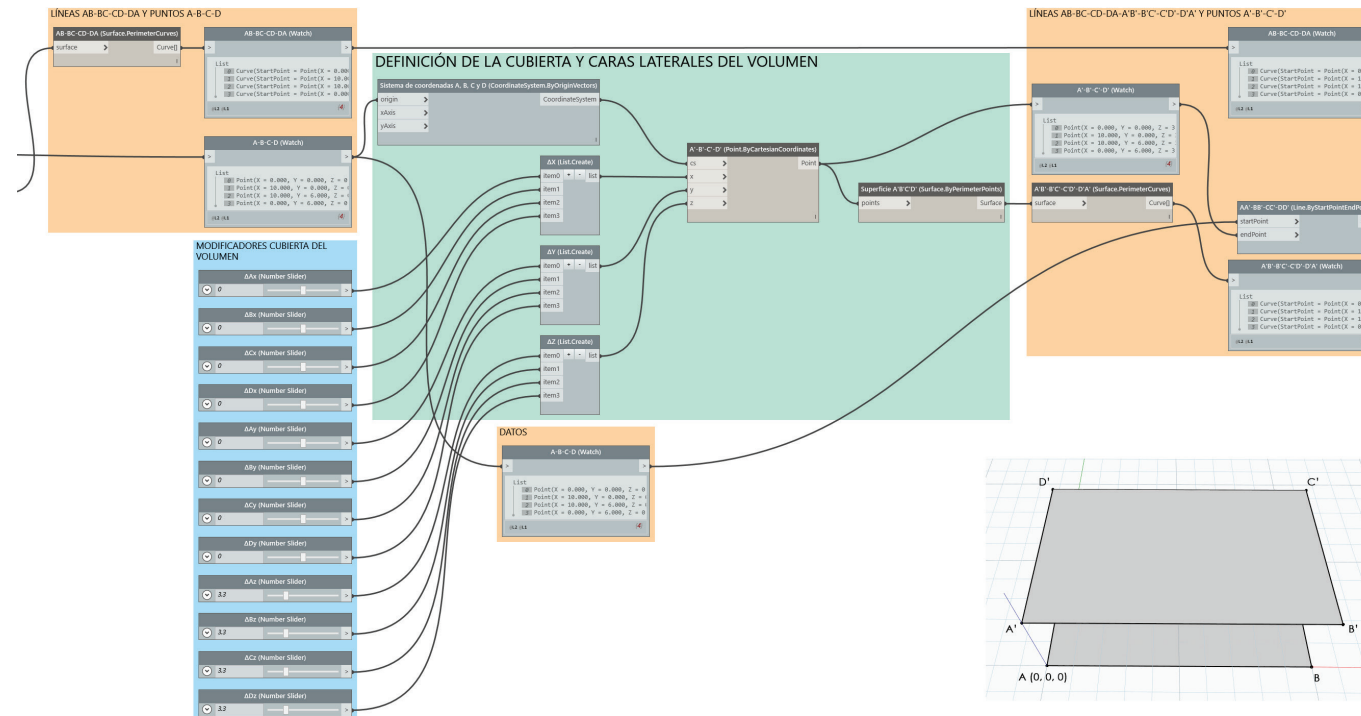


Imagen 2.22. Algoritmo que define la cubierta del volumen y sus modificadores - Captura de pantalla (Elaboración propia, 2016)

### 2.3.1.B. DEFINICIÓN DE LA CUBIERTA Y CARAS LATERALES DEL VOLUMEN

Los puntos a crearse para definir la cubierta tienen base en los puntos A-B-C-D, y deben ser capaces de moverse individualmente en cualquiera de los ejes x, y, z.

Los puntos A', B', C' y D' se crearán en la misma posición que los puntos A, B, C y D, por lo tanto, lo principal es determinar el desplazamiento en cada uno de los ejes que estos puntos presentan a partir de su punto original. Con este fin se crean los nodos "ΔAx (Number Slider)", "ΔAy (Number Slider)", "ΔAz (Number Slider)", "ΔBx (Number Slider)", "ΔBy (Number Slider)", "ΔBz (Number Slider)", "ΔCx (Number Slider)", "ΔCy (Number Slider)", "ΔCz (Number Slider)", "ΔDx (Number Slider)", "ΔDy (Number Slider)" y "ΔDz (Number Slider)", todos los nodos que refieren al eje x o y tienen un rango de -10 a 10 con un intervalo de 0.5, mientras que los nodos que refieren al eje z tienen un rango de 1 a 10 con intervalos de 0.5.

Es necesario crear un sistema de coordenadas en cada uno de los puntos A, B, C y D, para que los puntos A', B', C' y D' tengan estos sistemas de coordenadas como origen, para esto se crea el nodo "Sistema de coordenadas A, B, C y D (CoordinateSystem.ByOriginVectors)" y en la entrada origen conectamos la salida del nodo "A-B-C-D (Watch)", que contiene una lista con los puntos A, B, C y

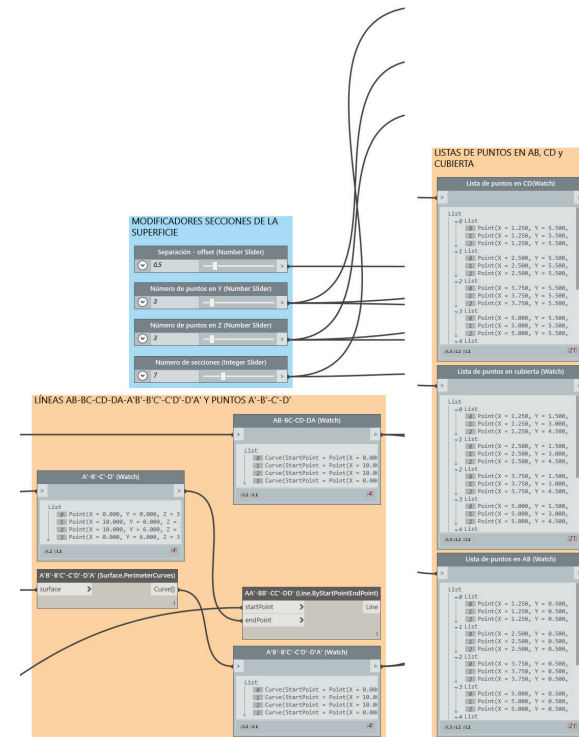
D, el resultado es una lista de sistemas de coordenadas cartesianas con origen en el listado de puntos A, B, C y D.

Después se necesitan crear listas que contengan los valores de desplazamiento de x, y, z, tres listas en total, con este fin se crean los nodos "ΔX (List.Create)", "ΔY (List.Create)" y "ΔZ (List.Create)", en la primera lista se conectan los valores de ΔAx, ΔBx, ΔCx y ΔDx, en la segunda lista se conectan los valores de ΔAy, ΔBy, ΔCy y ΔDy, en la última lista se colocan los valores de ΔAz, ΔBz, ΔCz y ΔDz.

Para crear los puntos A', B', C' y D' se ha creado el nodo "A'-B'-C'-D' (Point.ByCartesianCoordinates)", a las entradas de este nodo se conectan las salidas de los nodos "Sistema de coordenadas A, B, C y D (CoordinateSystem.ByOriginVectors)", "ΔX (List.Create)", "ΔY (List.Create)" y "ΔZ (List.Create)", el resultado es el listado de puntos A', B', C' y D', creados con los desplazamientos indicados en los nodos modificadores.

La salida del nodo "A'-B'-C'-D' (Point.ByCartesianCoordinates)" se conecta con la entrada del nodo "Superficie A'B'C'D' (Surface.ByPerimeterPoints)" para definir la superficie de la cubierta.

Para representar las líneas que delimitan la superficie A'B'C'D' se conecta la salida del nodo "Superficie A'B'C'D' (Surface.ByPerimeterPoints)" a la



**Imagen 2.23.** Modificadores y resultados del grupo que define las secciones que se crean en las caras del volumen - Captura de pantalla (Elaboración propia, 2016)

entrada del nodo "A'-B'-C'-C'D'-D'A' (Surface.PerimeterCurves)", mientras que el nodo "AA'-BB'-CC'-DD' (Line.ByStartPointEndPoint)" utiliza como entrada los listados de puntos de la base y la cubierta para definir las aristas verticales del volumen.

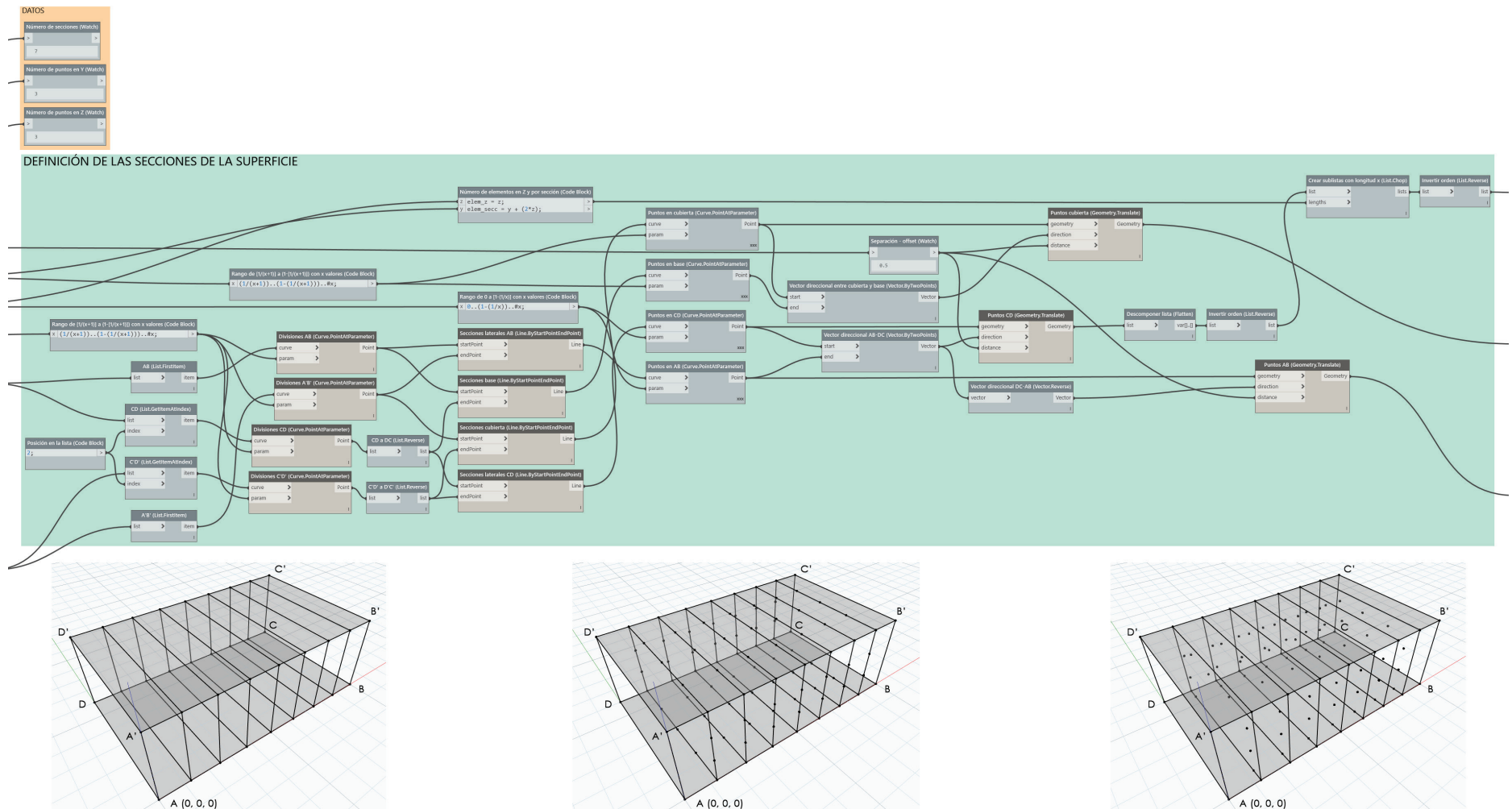


Imagen 2.24. Algoritmo que define las secciones que se crean en las caras del volumen - Captura de pantalla (Elaboración propia, 2016)



Los nodos "A'-B'-C'-D' (Watch)", "AB-BC-CD-DA (Watch)" y "A'B'-B'C'-C'D'-D'A' (Watch)", muestran los contenidos de las listas de puntos de la cubierta, lista de líneas que conforman la base y la lista de líneas que conforman la cubierta.

Nodos modificadores en esta sección:

- ΔAx (Number Slider)
- ΔAy (Number Slider)
- ΔAz (Number Slider)
- ΔBx (Number Slider)
- ΔBy (Number Slider)
- ΔBz (Number Slider)
- ΔCx (Number Slider)
- ΔCy (Number Slider)
- ΔCz (Number Slider)
- ΔDx (Number Slider)
- ΔDy (Number Slider)
- ΔDz (Number Slider)

Nodos resultados en esta sección:

- A'B'-B'C'-C'D'-D'A' (Surface. PerimeterCurves)

- A'-B'-C'-D' (Watch)
- AB-BC-CD-DA (Watch)
- AA'-BB'-CC'-DD' (Line. ByStartPointEndPoint)
- A'B'-B'C'-C'D'-D'A' (Watch)
- A-B-C-D (Watch)

### 2.3.1.c. DEFINICIÓN DE LAS SECCIONES DE LA SUPERFICIE

Habiendo definido toda la geometría necesaria para crear el volumen que va a contener al diseño, se procede a hacer las modificaciones en el volumen para generar el diseño.

Se crean los nodos "Separación – offset (Number Slider)" con un rango de 0.2 a 2 e incrementos de 0.1, "Número de puntos en Y (Number Slider)" con rango de 2 a 10 y con incrementos de 1, "Número de puntos en Z (Number Slider)" con rango de 2 a 10 e incrementos de 1, y "Número de secciones (Integer Slider)" con rango de 2 a 20 e incrementos de 1. Estos nodos permitirán la modificación de las variables que definen las secciones en el volumen.

Los nodos "AB (List.FirstItem)", "CD (List.GetItemAtIndex)", "C'D' (List.GetItemAtIndex)" y "A'B' (List.FirstItem)" extraen información de otros nodos, en el caso de los de tipo List.FirstItem extrae el

primer valor a partir de una lista, mientras que los de tipo List.GetItemAtIndex extrae el valor ubicado en la posición n de la lista, a n se le debe agregar un valor numérico, en este caso en particular necesitamos extraer el tercer valor de la lista, por lo que el número que se ingresa en el nodo "Posición en la lista (Code Block)" es el 2, esto porque en Dynamo las listas están numeradas a partir del número 0 en adelante, y 2 sería el tercer valor en la lista.

Los nodos anteriores definen las líneas AB, CD, A'B' y C'D'. Se desea crear una lista de puntos en cada una de estas líneas que contenga la cantidad de puntos según el parámetro ingresado en el nodo "Número de secciones (Integer Slider)", con este fin se utilizan los nodos "Divisiones AB (Curve.PointAtParameter)", "Divisiones A'B' (Curve.PointAtParameter)", "Divisiones CD (Curve.PointAtParameter)" y "Divisiones C'D' (Curve.PointAtParameter)", este tipo de nodo permite crear un punto o conjunto de puntos a lo largo de una curva a partir de la ubicación del punto en la longitud de la curva, para esto Dynamo interpreta como si la curva tuviera una longitud de uno, así la ubicación del punto se define mediante el ingreso de un valor numérico del 0 (inicio de la curva) al 1 (final de la curva).

Para determinar los lugares en los que se desea que se creen los puntos se crea el nodo "Rango de [1/(x+1)] a {1-[1/(x+1)]} con x valores (Code Block)", los nodos



de tipo Code Block permiten el ingreso de código para ejecutar acciones en el programa, si se incluye cualquier letra dentro de una fórmula en estos nodos, el programa inmediatamente la define como variable y crea una entrada para dicha variable, se establece x como variable, esta será igual al número de secciones.

Dentro del editor de código, se establece un valor inicial y uno final para un listado de valores, y se aplica escribiendo el valor inicial seguido de dos puntos (..) y el valor final de la lista, también se puede establecer los intervalos a los que se desea se creen nuevos elementos para la lista, o el número de elementos que conforma la lista, para el primer caso seguido del valor final de la lista se ingresan dos puntos (..) y el número que servirá de intervalo para nuevos valores, en el segundo caso seguido del valor final de la lista se colocan dos puntos (..) seguido de un símbolo de numeral (#) y el número de elementos que deben conformar la lista. Para terminar una línea de código se debe ingresar un punto y coma (;).

Se ingresa el siguiente código dentro del nodo:

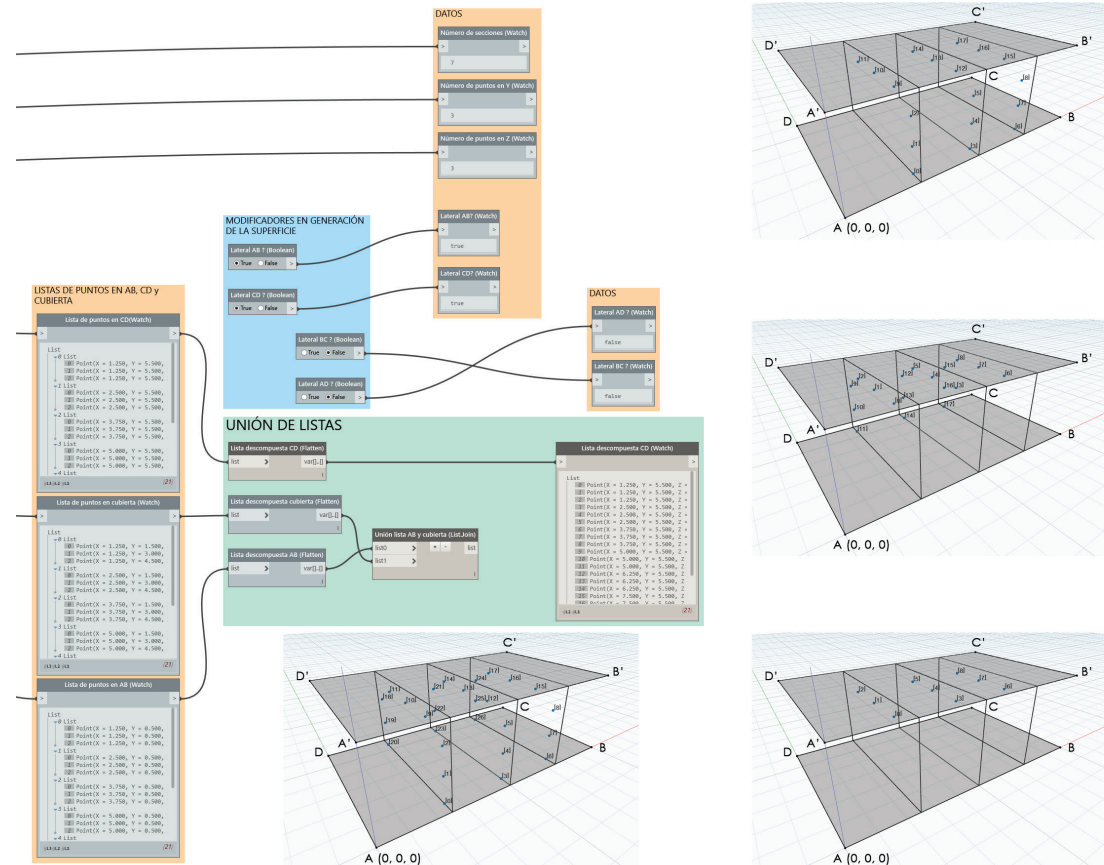
$(1/(x+1))..(1-(1/(x+1)))..#x;$

La línea de código crea una lista de valores que inicia en  $1/(x+1)$  y termina en  $1-[1/(x+1)]$ , con x número de valores. No se considera el inicio o el final de las líneas.

La salida de este nodo se conecta a la entrada de parámetro de los nodos "Divisiones AB (Curve.PointAtParameter)", "Divisiones A'B' (Curve.PointAtParameter)", "Divisiones CD (Curve.PointAtParameter)" y "Divisiones C'D' (Curve.PointAtParameter)", el resultado será un listado de x número de

puntos comprendidos entre los extremos de la línea para cada uno de los nodos, es decir cuatro listas en total.

Estas listas de puntos se usan para crear líneas que van desde AB a A'B', de A'B' a C'D', de C'D' a CD y de CD a AB.



**Imagen 2.25.** Algoritmo para unir las listas de puntos definidas en las secciones - Captura de pantalla (Elaboración propia, 2016)



Las listas de puntos de CD y C'D' deben ser invertidas, es decir el primer elemento debe volverse el último y viceversa, para mantener el mismo sentido que las listas de AB y A'B', para esto se crean los nodos "CD a DC (List.Reverse)" y "C'D' a D'C' (List.Reverse)".

Se crean los nodos "Secciones laterales AB (Line.ByStartPointEndPoint)", "Secciones laterales CD (Line.ByStartPointEndPoint)", "Secciones base (Line.ByStartPointEndPoint)" y "Secciones cubierta (Line.ByStartPointEndPoint)". A las secciones de la base se les conecta el listado de puntos de AB en los puntos iniciales y el listado de CD en los puntos finales, a las secciones de la cubierta se conecta el listado de puntos A'B' en los puntos iniciales y en los puntos finales el listado C'D', para las secciones laterales AB se conectan las listas AB en los puntos iniciales y A'B' en los finales, finalmente en las secciones laterales CD se conectan las listas CD en los puntos iniciales y C'D' en los finales.

Igual que con las líneas AB, CD, A'B' y C'D', las líneas creadas en el paso anterior deben ser divididas en partes iguales para colocar los puntos que servirán como puntos de control. Con este fin se crean los nodos "Puntos en AB (Curve.PointAtParameter)", "Puntos en CD (Curve.PointAtParameter)", "Puntos en base (Curve.PointAtParameter)" y "Puntos en cubierta (Curve.PointAtParameter)", en

las entradas de curva de estos nodos se conectan las salidas respectivas de los nodos Line.ByStartPointEndPoint.

En las líneas en base y cubierta no se desea considerar los puntos iniciales ni finales, por lo que se ocupa el mismo nodo "Rango de  $[1/(x+1)]$  a  $\{1-[1/(x+1)]\}$  con x valores (Code Block)", con la diferencia que la entrada de x se conecta con la salida del nodo "Número de puntos en Y (Number Slider)", el código se mantiene igual.

En las líneas que se encuentran en las laterales AB y CD se desea considerar solamente el punto inicial, más no el final, para esto se crea el nodo "Rango de 0 a  $[1-(1/x)]$  con x valores (Code Block)", el nodo debe definir una lista de valores que inicie en 0 y termine en  $1-(1/x)$ , y que contenga x número de valores, el código utilizado es el siguiente:

$0..(1-(1/x))..#x;$

Las listas de puntos creadas ahora necesitan desplazarse hacia el interior del volumen para así dar un margen de movimiento cuando se aplique la ecuación de la onda a los puntos. La dirección en la que los puntos se deben mover está dada por el vector comprendido entre los puntos de caras opuestas, es decir, un conjunto de vectores entre los puntos de cubierta y base, y otro conjunto de vectores entre los puntos de las laterales AB y CD. Hay que tener en cuenta la dirección de los

vectores, pues se necesita que los puntos se muevan solamente hacia el interior del volumen.

Se crean los nodos "Vector direccional entre cubierta y base (Vector.ByTwoPoints)" y "Vector direccional AB-DC (Vector.ByTwoPoints)", estos nodos necesitan como entradas un punto de inicio y uno final para definir la dirección de los vectores.

Se conecta entonces en la entrada de inicio del nodo para cubierta y base la salida del nodo que define los puntos en cubierta, y en la entrada de final se conecta la salida del nodo que define los puntos en base, de esta manera el vector resultante tendrá dirección hacia el interior del volumen.

En el nodo que determina el vector para AB-DC se conecta a la entrada de inicio la salida del nodo que define los puntos en la lateral AB, y como entrada de final se conecta la salida del nodo que define los puntos para la lateral CD, la dirección de los vectores resultantes de este nodo es de AB hacia DC, ahora se necesita duplicar estos vectores e invertirlos para poder aplicarlos a la cara CD, para esto se crea el nodo "Vector direccional DC-AB (Vector.Reverse)" y se conecta la entrada con la salida de los vectores resultantes de AB a DC.

Para mover los puntos hacia el interior del volumen se crean los nodos "Puntos cubierta (Geometry.Translate)", "Puntos





AB (Geometry.Translate)" y "Puntos CD (Geometry.Translate)", a las entradas de geometría se conectan los elementos que se quieren mover, es decir los puntos de la cubierta, los puntos del lateral AB y los puntos del lateral CD, a las entradas de dirección se conectan los vectores que hemos definido, teniendo en cuenta que el desplazamiento se realice hacia el interior del volumen, y para las entradas de distancia se conecta la salida del nodo "Separación – offset (Number Slider)", este valor determinará la distancia que los puntos se separarán de las caras en las cuales se encontraban originalmente.

Los listados resultantes son los puntos en la lateral AB, lateral CD y cubierta, ya aplicado el desplazamiento, la lista de puntos en CD debe ser modificada antes de poder utilizarse con el resto de listas, los puntos en AB se encuentran en el sentido de AB a A'B', los puntos de la cubierta están en el sentido de A'B' a C'D', y los puntos de CD se encuentran en sentido de CD a C'D', es decir en sentido contrario a las otras listas, por eso se necesita invertir el orden de la lista. Primero se conecta la salida del nodo "Puntos CD (Geometry.Translate)" a la entrada del nodo "Descomponer lista (Flatten)", se utiliza este tipo de nodo para "aplanar" a un solo nivel una lista cuando contiene sub-listas, luego se conecta la salida de este nodo a la entrada del nodo "Invertir orden (List.Reverse)" que invertirá el sentido de la lista. Ahora la lista se encuentra en el

mismo sentido que las otras listas, pero se debe mantener la estructura original de lista y sub-listas, para esto se utiliza el nodo "Crear sublistas con longitud x (List.Chop)", este tipo de nodo permite dividir una lista en sub-listas que contengan un número de elementos establecido por la entrada longitudes, cada sub-lista debe contener un número igual al número de puntos creados en Z, se conecta entonces la salida del nodo "Número de puntos en Z (Number Slider)". Finalmente se vuelve a invertir el orden de la lista para que esté acorde a los otros puntos mediante el nodo "Invertir orden (List.Reverse)".

Los resultados de este grupo de procesos son los nodos "Lista de puntos en AB (Watch)", "Lista de puntos en CD (Watch)" y "Lista de puntos en cubierta (Watch)", las listas de las laterales AB y CD están comprendidas de un número de sub-listas igual al número de secciones y cada sub-lista contiene un número de elementos igual al número de puntos en Z, mientras que la lista de la cubierta tiene un número de sub-listas igual al número de secciones pero cada sub-lista está compuesta de un número de elementos igual al número de puntos en Y.

Nodos modificadores en esta sección:

- Separación – offset (Number Slider)
- Número de puntos en Y (Number Slider)

- Número de puntos en Z (Number Slider)
- Número de secciones (Integer Slider)

Nodos resultados en esta sección:

- Lista de puntos en AB (Watch)
- Lista de puntos en CD(Watch)
- Lista de puntos en cubierta (Watch)
- Número de puntos en Y (Watch)
- Número de puntos en Z (Watch)
- Número de secciones (Watch)

### 2.3.1.D. UNIÓN DE LISTAS

Tras haber definido los puntos que conformarán las secciones debemos ordenarlos de tal manera que, al unir las tres listas en una sola, los puntos de una misma sección se encuentren en una sub-lista y así para todas las secciones.

Para maximizar la capacidad de exploración y de experimentación del algoritmo se han agregado unas variables que permitan considerar o excluir caras diferentes del volumen para el cómputo final de los puntos y creación de secciones, estas variables son de carácter booleano y se definen mediante la selección entre dos opciones, verdadero o falso, 0 o 1, se crean los nodos "Lateral AB ? (Boolean)", Lateral

CD ? (Boolean)", "Lateral BC ? (Boolean)" y "Lateral AD ? (Boolean)", cada uno de los mismos permitirá seleccionar si se quiere considerar o no la cara específica en el cálculo final de las secciones.

Para esto se aplican dos grupos de procesos condicionales, que determinan si se cumple una condición y dan como resultado un proceso específico, sin no se cumple la condición se ejecuta otra rama de procesos.

Para poder trabajar con las listas es primero necesario aplanarla hasta que contenga un solo nivel, para esto se utilizan los nodos "Lista descompuesta AB (Flatten)",

"Lista descompuesta CD (Flatten)" y "Lista descompuesta cubierta (Flatten)", ahora cada lista contiene un solo nivel.

Se crea el nodo "Unión lista AB y cubierta (List.Join)", este nodo une en una sola lista los elementos de las listas de puntos en la lateral AB y puntos en cubierta.

### Condicional lateral AB

Lo que se pretende definir en este proceso es que lista ocupar en la continuación de ejecución del algoritmo, la primera opción es solamente la lista de puntos en cubierta y la segunda opción es la lista obtenida al unir la lista de cubierta y la lista del lateral AB. Si no se incluyen los puntos del lateral

AB el resultado de las secciones obviará ese lado para la definición de la superficie.

Como primer paso se crea el nodo "Listas a elegir (List.Create)", se le establece dos entradas, lo que hará este nodo es crear una lista con dos elementos, cada elemento a la vez mantendrá la estructura original de lista que tenía antes de conectarse al nodo, se conecta a la entrada item0 la lista resultante del nodo "Unión lista AB y cubierta (List.Join)" y en la entrada item1 se conecta la lista resultado del nodo "Lista descompuesta cubierta (Flatten)". Después se necesita evaluar el número de elementos que contiene cada una de las sub-listas, con este fin se crea

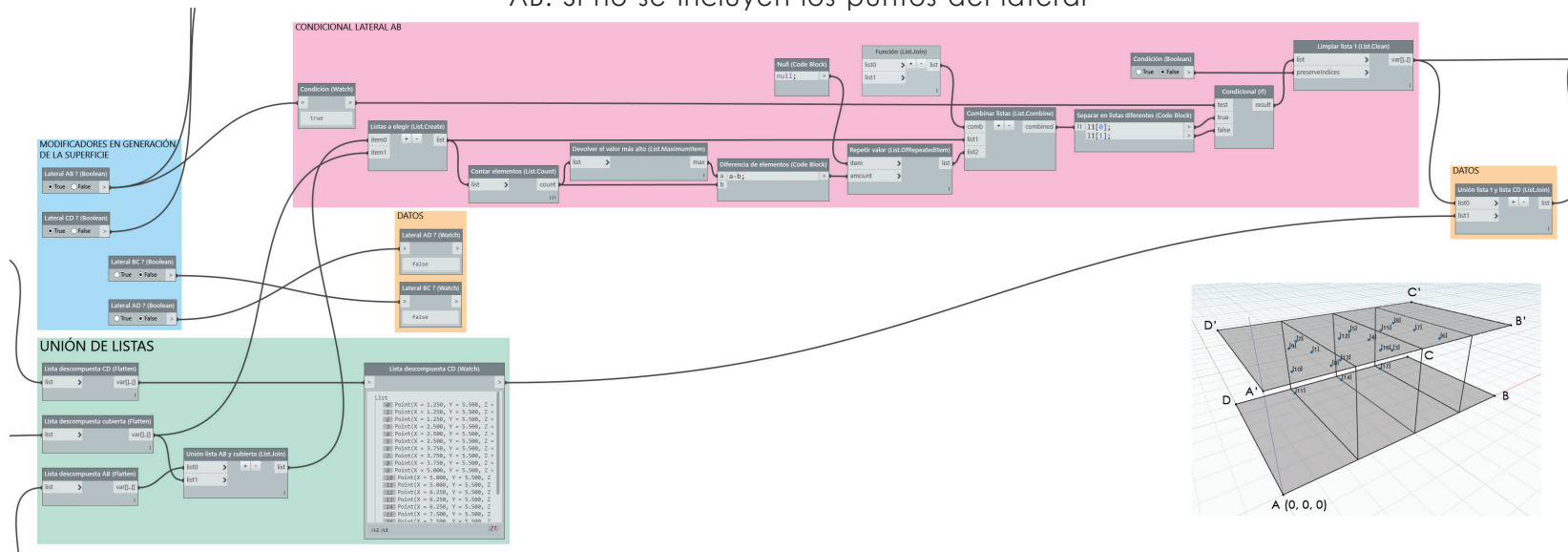
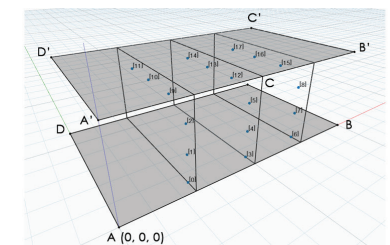
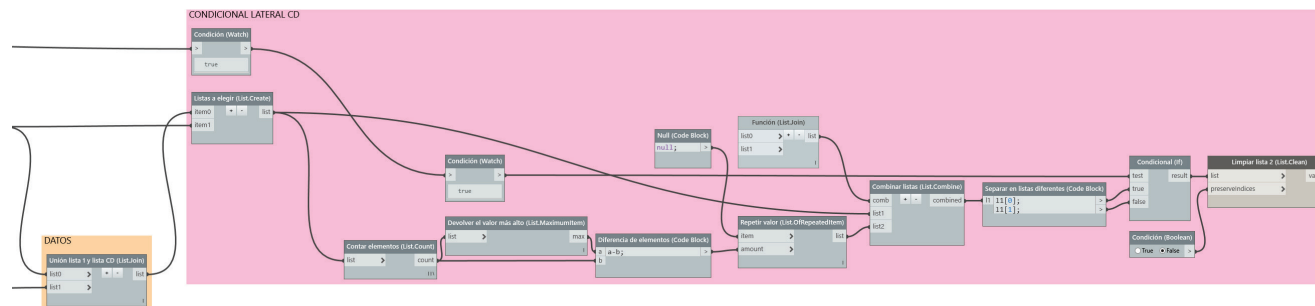


Imagen 2.26. Algoritmo que determina la incorporación de los puntos del lateral AB al listado final - Captura de pantalla (Elaboración propia, 2016)

Esto crea una entrada para a y una para b, en la entrada de a colocamos el resultado del valor más alto, mientras que en la entrada de b conectamos el resultado del conteo de elementos en la lista. Si,

Ahora se debe crear una lista que a partir del número de elementos resultante de la resta anterior, cree ese número de veces un elemento determinado, para el ejemplo anterior sería dos listas, la primera con cero elementos y la segunda con  $m-n$  número de elementos, para esto creamos el nodo "Repetir valor (List.OfRepeatedItem)" y a la entrada de cantidad conectamos el resultado del nodo "Diferencia de elementos (Code Block)", el elemento a

El resultado de esta operación será una lista con dos sub-listas, la primera una lista vacía, pues no contiene elementos, y la segunda una lista con elementos null. El próximo paso es combinar esta lista con la lista del nodo "Listas a elegir (List. Create)", se requiere que a los elementos de la primera sub-lista de la lista original se añadan los elementos de la primera sub-lista de valores null, lo mismo para la segunda sub-lista, para esto se crea el nodo "Combinar listas (List.Combine)" y se ingresan las dos listas a sus entradas, la entrada comb requiere definir el criterio que se va a aplicar para combinar las listas, se necesita que la unión de dos sub-listas sea una sola, por lo que se le ingresa el nodo "Función (List.Join)".



CAPÍTULO II - DISEÑO DE UN ESPACIO MEDIANTE EL USO DE ALGORITMOS GENERATIVOS



En el nodo se crea la entrada l1, a esta se conecta el resultado de la combinación de las listas, las salidas son dos listas, la una contiene solamente puntos (unión de los puntos de cubierta y puntos en AB), mientras que la otra contiene puntos y elementos null (solamente los puntos de la cubierta), las dos listas contienen el mismo número de elementos, que es una condición del programa para poder utilizar una condicional cuando se trabaja con listas.

A esta altura ya se puede ingresar la condicional para determinar cuál de las dos listas ocupar, se crea el nodo "Condicional (If)", este posee una entrada para los datos que utilizará si el valor del booleano es verdadero y otra entrada para los datos que se utilizarán si el valor del booleano es falso, además posee la entrada para ingresar el booleano. A la entrada test se conecta el resultado del nodo "Lateral AB ? (Boolean)", a la entrada verdadero se conecta la lista l1[0] y a la entrada falso se conecta la lista l1[1].

Finalmente, si se termina ocupando la lista que contiene elementos null estos van a generar problemas en los procesos del programa, por lo que es necesario limpiar la lista de todos estos valores, para esto se crea el nodo "Limpiar lista 1 (List.Clean)", este nodo elimina todos los valores null y listas vacías que existan en la lista que se ingrese, se ocupa el nodo "Condición (Boolean)" con la opción falso

seleccionada para determinar que no se preserven los índices originales de la lista.

El resultado de este proceso es una lista que contiene los puntos de AB y la cubierta, si el nodo "Lateral AB ? (Boolean)" tiene un valor verdadero, mientras que su valor es falso, la lista solamente contendrá los puntos de la cubierta.

### Condicional lateral CD

Básicamente se repite todo lo realizado para la condicional del lateral AB, pero se sustituyen los elementos item0 e item1 y se ocupa la variable booleana del nodo "Lateral CD ? (Boolean)" en lugar del nodo "Lateral AB ? (Boolean)". Para la entrada item0 del nodo "Listas a elegir (List.Create)" se utiliza la salida del nodo "Limpiar lista 1 (List.Clean)" y para la entrada item1 se utiliza el resultado del nodo "Unión lista 1 y lista CD (List.Join)", esta lista se crea al unir las salidas de los nodos "Limpiar lista 1 (List.Clean)" y "Lista descompuesta CD (Flatten)".

El resultado final de este grupo de procesos es el nodo "Limpiar lista 2 (List.Clean)" que contiene los puntos que se utilizarán para definir las secciones. En este punto hay cuatro resultados posibles, de acuerdo a los valores de las variables definidas en los nodos "Lateral AB ? (Boolean)" y "Lateral CD ? (Boolean)", la lista puede ser uno de estos cuatro casos:

- Variable AB, verdadero, variable CD, verdadero, resultado, la lista contiene los puntos de las laterales AB y CD más los puntos de la cubierta.
- Variable AB, falso, variable CD, verdadero, resultado, la lista contiene los puntos de la lateral CD más los puntos de la cubierta.
- Variable AB, verdadero, variable CD, falso, resultado, la lista contiene los puntos de la lateral AB más los puntos de la cubierta.
- Variable AB, falso, variable CD, falso, resultado, la lista contiene solamente los puntos de la cubierta.

Nodos modificadores en esta sección:

- Lateral AB ? (Boolean)
- Lateral CD ? (Boolean)
- Lateral BC ? (Boolean)
- Lateral AD ? (Boolean)

Nodos resultados en esta sección:

- Unión lista 1 y lista CD (List.Join)
- Lateral AB ? (Watch)
- Lateral CD ? (Watch)





### 2.3.1.E. SECCIONES FINALES

Una vez definida la lista con los puntos que van a definir las secciones, se necesita ordenar los puntos de tal manera que los que pertenecen a una misma sección se encuentren en la misma sub-lista.

Se debe establecer el número de elementos que conforman la lista total, ya que hay cuatro posibilidades diferentes de creación de la lista, el número de elementos que contiene cada sección y el número de elementos que contiene cada cara del volumen.

Se crea el nodo "Condiciones para dividir las listas (Code Block)" y se ingresa el siguiente código:

```
0;  
n*z;  
n*y;  
(n*z)+(n*y);  
(n*z)+(n*y)+(n*z);
```

Se crean en el nodo las entradas n, z, y, en "n" se ingresa el número de secciones, en "z" se ingresa el valor del número de puntos en z y en "y" se ingresa el valor del número de puntos en y. Se conectan a los nodos "Número de secciones (Watch)", "Número de puntos en Z (Watch)" y "Número de puntos en Y (Watch)".

El código define cinco salidas diferentes, la primera el valor 0, la segunda el producto del número de secciones por el número de puntos en z, la tercera el producto del número de secciones por el número de puntos en y, la cuarta la suma de la segunda y la tercera salida, y la quinta la suma de dos veces el valor de la segunda salida más el valor de la tercera salida.

Estos valores permitirán dividir la lista completa de puntos en sub-listas que contengan los puntos de una misma cara del volumen. Se crean los nodos "Dividir lista 1 (List.Slice)", "Dividir lista 2 (List.Slice)" y "Dividir lista 3 (List.Slice)".

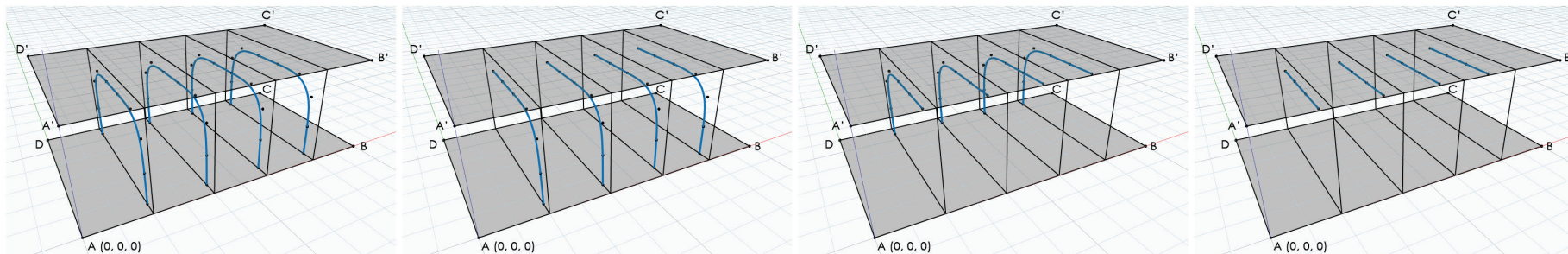
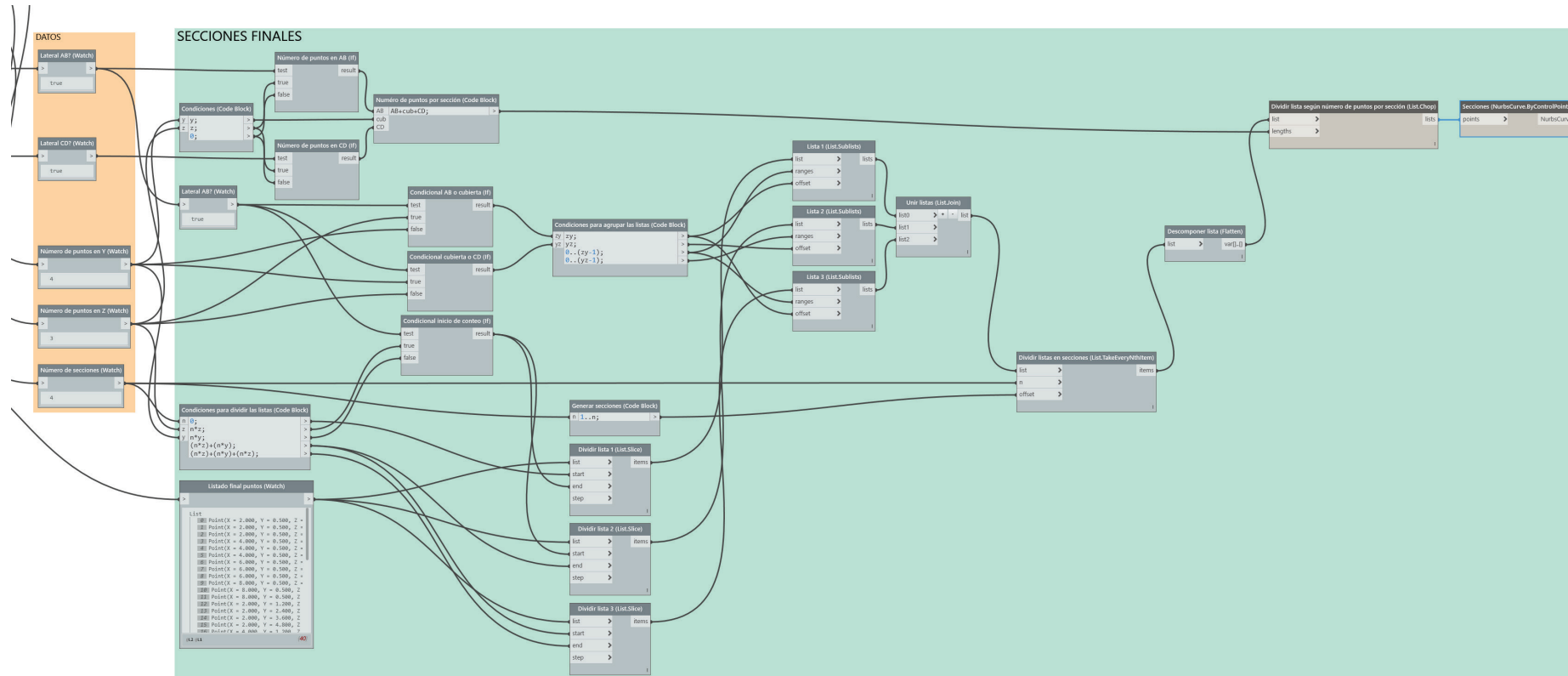
En el nodo "Dividir lista 1 (List.Slice)" se conecta la entrada lista a la lista total de puntos del nodo "Listado final puntos (Watch)", el tipo de nodo List.Slice permite seccionar una lista desde un valor de índice inicial a un índice final, para crear una nueva lista con solamente esos elementos.

Se define el fin de la lista con el nodo "Condicional inicio de conteo (If)", en la entrada test se conecta el resultado del nodo "Lateral AB ? (Watch)", en la entrada verdadero se conecta la segunda salida del nodo "Condiciones para dividir las listas (Code Block)", es decir el número total de puntos en las caras laterales, y a la entrada falso se conecta la tercera salida del nodo "Condiciones para dividir las listas (Code Block)", es decir el número total de puntos en la cubierta.

En el nodo "Dividir lista 1 (List.Slice)" se ingresa como índice inicial la primera salida del nodo "Condiciones para dividir las listas (Code Block)", es decir el valor de 0 y para el índice final se conecta el resultado del nodo "Condicional inicio de conteo (If)", de esta manera el resultado de este nodo será una lista con los puntos del lateral AB, si la variable AB es verdadero, si es falso, la lista contendrá los puntos de la cubierta.

Se crea el nodo "Dividir lista 2 (List.Slice)", se conecta el resultado del nodo "Listado final puntos (Watch)", al índice inicial se le conecta el resultado del nodo "Condicional inicio de conteo (If)" y al índice final se le conecta la cuarta salida del nodo "Condiciones para dividir las listas (Code Block)", siendo así el resultado de esta lista será una lista en la que, si el valor de la variable AB es verdadero, contendrá los puntos de la cubierta, si la variable AB es falso, entonces la lista contendrá los puntos de la lateral CD y en el caso de que ambas variables, AB y CD, sean falso, el resultado será una lista vacía.

Por último, se crea el nodo "Dividir lista 3 (List.Slice)", de igual manera se conecta la lista resultado del nodo "Listado final puntos (Watch)", como índice de inicio se utiliza la cuarta salida del nodo "Condiciones para dividir las listas (Code Block)" y como índice final se ocupa la quinta salida del mismo nodo. En el caso de que las variables AB y CD sean verdadero, el resultado será un listado que contiene



**Imagen 2.28.** Algoritmo para crear las secciones finales - Captura de pantalla (Elaboración propia, 2016)



los puntos de la lateral CD, en caso de que cualquiera de las variables, AB o CD, sea falso, el resultado será una lista vacía.

Tras haber definido estas tres listas, el siguiente paso es dividir cada una en un número de sub-listas igual al número de secciones, cada sub-lista contendrá solo los puntos que pertenezcan a una misma sección, para esto se crean los nodos "Lista 1 (List.Sublists)", "Lista 2 (List.Sublists)" y "Lista 3 (List.Sublists)", este tipo de nodo crea sub-listas a partir de una lista, definidas por un rango y un desfase.

Además, se crean los nodos "Condicional AB o cubierta (If)", "Condicional cubierta o CD (If)" y "Condiciones para agrupar las listas (Code Block)". En el nodo "Condicional AB o cubierta (If)" se conecta el resultado del nodo "Lateral AB ? (Watch)" en la entrada test, para el valor de verdadero se asigna el resultado del nodo "Número de puntos en Z (Watch)" y para el valor de falso se ingresa el resultado del nodo "Número de puntos en Y (Watch)", de esta manera si el valor de la variable AB es verdadero, el resultado será igual al número de puntos en Z, mientras que, si es falso, el resultado será el número de puntos en Y. En el nodo "Condicional cubierta o CD (If)" se conecta a la entrada test el resultado del nodo "Lateral CD ? (Watch)", en este caso se conecta el valor de verdadero al resultado del nodo "Número de puntos en Y (Watch)" y al valor de falso se conecta el resultado del nodo "Número de puntos en

Z (Watch)", es decir que, si la variable CD tiene el valor de verdadero, el resultado será igual al número de puntos en Y, mientras que, si es falso, el resultado será igual al número de puntos en Z. En el nodo "Condiciones para agrupar las listas (Code Block)" se ingresa el siguiente código:

```
zy;  
yz;  
0..(zy-1);  
0..(yz-1);
```

Mediante esto se crean las variables zy y yz, la primera se conecta al resultado del nodo "Condicional AB o cubierta (If)", mientras que la segunda se conecta al resultado del nodo "Condicional cubierta o CD (If)", el texto 0..(zy-1) crea un rango que inicia en 0 y termina en el valor de zy menos uno, lo mismo se aplica para el texto 0..(yz-1), se crean así rangos que van de 0 al número de puntos en AB, número de puntos en cubierta o número de puntos en CD, todos menos uno, esto es así porque las listas en Dynamo empiezan con el índice de valor 0, es decir, una lista que contiene 3 elementos por ejemplo, tiene los índices [0], [1] y [2], siendo el último índice menor al número de elementos en uno.

Se conecta el resultado del nodo "Dividir lista 1 (List.Slice)" a la entrada lista del nodo "Lista 1 (List.Sublists)", la entrada rango se conecta con la tercera salida y

la entrada desfase con la primera salida del nodo "Condiciones para agrupar las listas (Code Block)". Se realiza algo similar con el nodo "Lista 2 (List.Sublists)", a la entrada de lista se conecta el resultado del nodo "Dividir lista 2 (List.Slice)", se utiliza como rango la cuarta salida del nodo "Condiciones para dividir las listas (Code Block)" y como desfase la segunda salida del mismo nodo. En el nodo "Lista 3 (List.Sublists)" se conecta la entrada lista al resultado del nodo "Dividir lista 3 (List.Slice)", como rango se toma la tercera salida y como desfase la primera salida del nodo "Condiciones para agrupar las listas (Code Block)".

El siguiente paso es unir las listas para poder luego agrupar las sub-listas de cada lista por sección. Según los valores de las variables AB y CD, pueden existir listas vacías, esto no presenta ningún inconveniente pues al unir una lista con otra lista que se encuentra vacía no se añade nada a la primera lista, lo cual resuelve cualquier posible error debido a esto.

Para unir las listas se crea el nodo "Unir listas (List.Join)", que contiene las entradas list0, list1 y list2. Se ingresa a list0 el resultado del nodo "Lista 1 (List.Sublists)", a list1 el resultado del nodo "Lista 2 (List.Sublists)" y a list2 el resultado del nodo "Lista 3 (List.Sublist)", el resultado será una sola lista que contiene sub-listas por cada sección.



Para dividir la lista que se acaba de crear se utiliza el nodo "Dividir listas en secciones (List.TakeEveryNthItem)", este nodo permite extraer los índices que sean múltiplos de un número ingresado, también se puede añadir un desfase. En la entrada lista del nodo añadimos el resultado del nodo "Unir listas (List.Join)", a la entrada n, los múltiplos de este número serán extraídos por el programa, se asigna el resultado del nodo "Número de secciones (Watch)", para definir el desfase se utiliza el nodo "Generar secciones (Code Block)" que contiene el siguiente código:

```
1..n;
```

Esto genera un listado de valores que empiezan en uno y terminan en n, con incrementos de uno, a la entrada n que se ha creado se ingresa el resultado del nodo "Número de secciones (Watch)".

El resultado del nodo "Dividir listas en secciones (List.TakeEveryNthItem)" es una lista de tres niveles, el primero contiene un número de elementos igual al número de secciones, cada una de estas listas está comprendida por sub-listas, una sub-lista si las variables AB y CD son falsas, dos sub-listas si una de las dos variables es falsa y la otra verdadera, y tres sub-listas en caso de que las dos variables sean verdaderas; a su vez cada sub-lista contiene una lista de puntos. Tras este paso, todos los puntos están ordenados de manera adecuada para poder generar las curvas.

Para poder generar las curvas hace falta solamente ordenar los puntos en listados en los cuales estén los puntos de una misma sección, con un número de listas igual al número de secciones.

Se utiliza el nodo "Descomponer lista (Flatten)" para aplanar los niveles de la lista resultante del nodo "Dividir listas en secciones (List.TakeEveryNthItem)", el resultado de este proceso se ingresa a la entrada lista del nodo "Dividir lista según número de puntos por sección (List.Chop)", y la longitud de las listas que se generan con este nodo es igual al resultado del nodo "Número de puntos por sección (Code Block)", el mismo que contiene el siguiente código:

```
AB+cub+CD;
```

Se crean las variables AB, cub y CD, además se crea el nodo "Condiciones (Code Block)" y se ingresa el siguiente código:

```
y;
```

```
z;
```

```
0;
```

A la variable y se conecta el resultado del nodo "Número de puntos en Y (Watch)", al a variable z se le asigna el resultado del nodo "Número de puntos en Z (Watch)". Se crean los nodos "Número de puntos en AB (If)" y "Número de puntos en CD (If)".

A la variable y se conecta el resultado del nodo "Número de puntos en Y (Watch)", al a variable z se le asigna el resultado del nodo "Número de puntos en Z (Watch)". Se crean los nodos "Número de puntos en AB (If)" y "Número de puntos en CD (If)".

En el nodo "Número de puntos en AB (If)" se conecta la entrada test al resultado del nodo "Lateral AB ? (Watch)", en la entrada verdadero se conecta con la segunda salida del nodo "Condiciones (Code Block)" y en la entrada falso, con la tercera salida. Se hace lo mismo para el nodo "Número de puntos en CD (If)" cambiando solamente la entrada test, ingresando aquí el resultado del nodo "Lateral CD ? (Watch)".

Dentro del nodo "Número de puntos por sección (Code Block)" se ingresa a la entrada AB el resultado del nodo "Número de puntos en AB (If)", la entrada cub se conecta con la primera salida del nodo "Condiciones (Code Block)" y la entrada CD se ingresa el resultado del nodo "Número de puntos en CD (If)". Este proceso permite definir el número de puntos que contiene cada sección para los diferentes valores y combinaciones de las variables AB y CD.

Finalmente, el resultado del nodo "Dividir lista según número de puntos por sección (List.Chop)" es un listado que contiene un número de sub-listas igual al número de secciones, y cada sub-lista contiene los puntos de una sección solamente.

El último paso es crear el nodo “Secciones (NurbsCurve.ByControlPoints)”, que genera finalmente elementos geométricos que representen las secciones, y que tiene como puntos de control los puntos definidos en el proceso anterior, se conecta la entrada puntos al resultado del nodo “Dividir lista según número de puntos por sección (List.Chop)”.

### 2.3.1.F. DEFINICIÓN DE LATERALES AD Y BC

Mediante esta consideración se incrementa en gran manera la versatilidad del algoritmo generador, puesto que al considerar e incluir los lados AD y BC a las secciones se puede conseguir diferentes tipologías de superficies, si se agrega al lado AD a las secciones, el

resultado será una superficie cerrada en el extremo inicial, mientras que si se añade al lado BC, la superficie será cerrada en el extremo final, si en cambio se añaden los dos lados a las secciones, el resultado es una superficie completamente cerrada en todos los extremos, finalmente si no se consideran estos lados para el cálculo de la superficie, el resultado es una superficie abierta en ambos extremos, el inicial y final.

Primero es necesario definir los segmentos AD y BC, para luego mediante una variable booleana establecer si se utiliza en conjunto con las demás secciones para generar la superficie.

Se crea el nodo “A-B-C-D (Watch)”, este nodo extrae los puntos que conforman

los vértices de la base del volumen, se conecta la salida de este nodo a la entrada del nodo “Separar puntos en listas diferentes (Code Block)”, este nodo tiene el siguiente código:

```
Puntos_base[0];
```

```
Puntos_base[3];
```

```
Puntos_base[1];
```

```
Puntos_base[2];
```

El código definido permite extraer el primer elemento de la lista, seguido por el cuarto elemento, segundo elemento y tercer elemento, en ese orden, en este caso el primer elemento es el punto A, el cuarto elemento es el punto D, el segundo elemento es el punto B y el tercer elemento es el punto C.

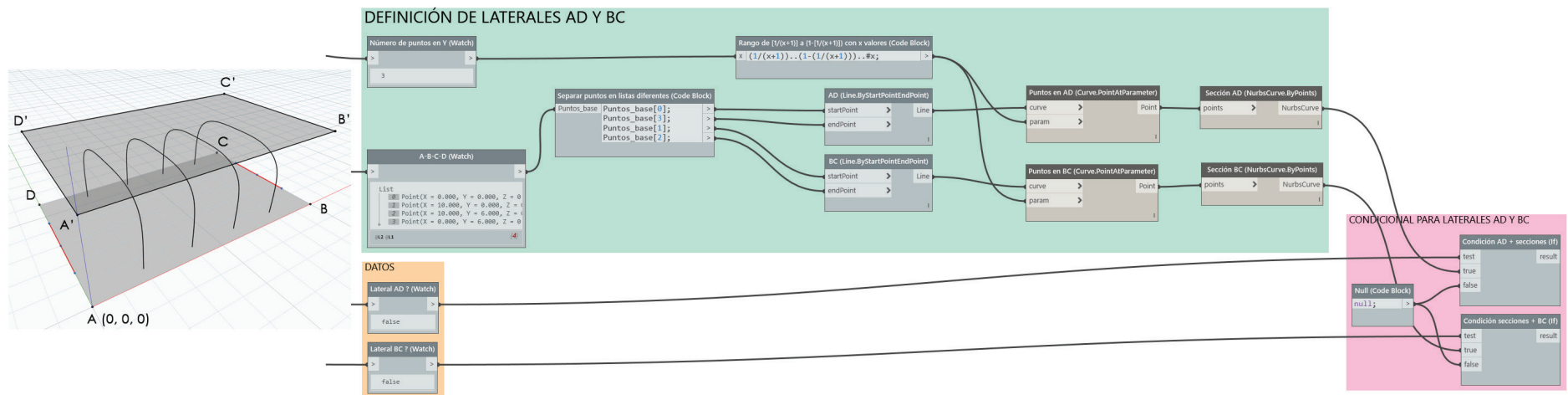


Imagen 2.29. Algoritmo que determina la incorporación de las laterales AD y BC a las secciones para generar la superficie - Captura de pantalla (Elaboración propia, 2016)



Se crean los nodos "AD (Line.ByStartPointEndPoint)" y "BC (Line.ByStartPointEndPoint)", para definir el lado AD se unen los puntos A y D, se conecta como punto inicial A y como punto final D, en BC se conecta como punto inicial B y como punto final C.

Ahora se deben dividir las líneas resultantes de la misma manera que se dividieron las líneas comprendidas entre A'B' y C'D', se utiliza el mismo rango, definido por el nodo "Rango de  $[1/(x+1)]$  a  $\{1-[1/(x+1)]\}$  con x valores (Code Block)" que contiene el siguiente código:

```
(1/(x+1))..(1-(1/(x+1)))..#x;
```

Este rango se conecta a la entrada parámetro de los nodos "Puntos en AD (Curve.PointAtParameter)" y "Puntos en BC (Curve.PointAtParameter)", y en las entradas de curva, se ingresan los resultados de "AD (Line.ByStartPointEndPoint)" y "BC (Line.ByStartPointEndPoint)" respectivamente.

El resultado es una lista de puntos para AD y otra para BC, utilizándolos como puntos de control se crean curvas NURBS, para tener el mismo tipo de curva que en las secciones, se crea entonces los nodos "Sección AD (NurbsCurve.ByPoints)" y "Sección BC (NurbsCurve.ByPoints)", a cada uno de estos nodos se les conecta la lista de puntos respectiva.

### Condición para laterales AD y BC

En lugar de determinar si se incorporan las laterales AD y BC en la lista de secciones, se van a incorporar directamente y se controla su intervención en la superficie reemplazando el valor de la lateral, AD o BC, por un valor null en caso de que el valor de las variables sea falso, en caso de ser verdadero, se utiliza directamente las laterales AD y BC.

Para esto se define el nodo "Null (Code Block)" con el siguiente código:

```
null;
```

Además, se crean los nodos "Condición AD + secciones (If)" y "Condición secciones + BC (If)", en el primer nodo se utiliza el resultado del nodo "Sección AD (NurbsCurve.ByPoints)" en la entrada de valor verdadero, y "Null (Code Block)" en la entrada de valor falso, en la entrada test se ingresa el resultado del nodo "Lateral AD ? (Watch)", en el nodo "Condición secciones + BC (If)" se conecta a la entrada de valor verdadero el resultado del nodo "Sección BC (NurbsCurve.ByPoints)", a la entrada de valor falso se conecta el resultado de "Null (Code Block)" y a la entrada test se conecta el resultado del nodo "Lateral BC ? (Watch)". Dependiendo del valor de las variables AD y BC, el resultado puede ser la línea AD, la línea BC o valores null, estos resultados se unirán directamente con la lista de las secciones.

### 2.3.1.G. VERIFICACIÓN Y CORRECCIÓN DE LA DIRECCIÓN DE LOS VECTORES NORMALES

Una vez que se han definido todas las secciones, se debe dividir cada una de estas secciones en partes iguales, para así crear un número determinado de puntos a lo largo de cada curva, estos puntos serán desplazados posteriormente hacia el interior o exterior del volumen para adaptarse a la curva generada por la ecuación de la onda.

Se crea entonces el nodo "Valores del 0 al 1 - x número de valores (Code Block)" y se ingresa el siguiente código:

```
0..1..#x;
```

De esta manera se genera un rango, definido entre 0 y 1, con un número de valores definido por la variable x, a esta entrada se conecta el resultado del nodo "Número de divisiones por sección (Integer Slider)", que permite modificar fácilmente la cantidad de puntos que se generen por sección, se debe considerar que mientras más puntos se creen, el resultado de la interpolación final de la curva va a ser mucho más definido, pero el tiempo de cómputo va a ser mayor que con menos puntos.

Con el mismo rango se procede a determinar el vector normal en cada uno de estos puntos sobre las curvas.

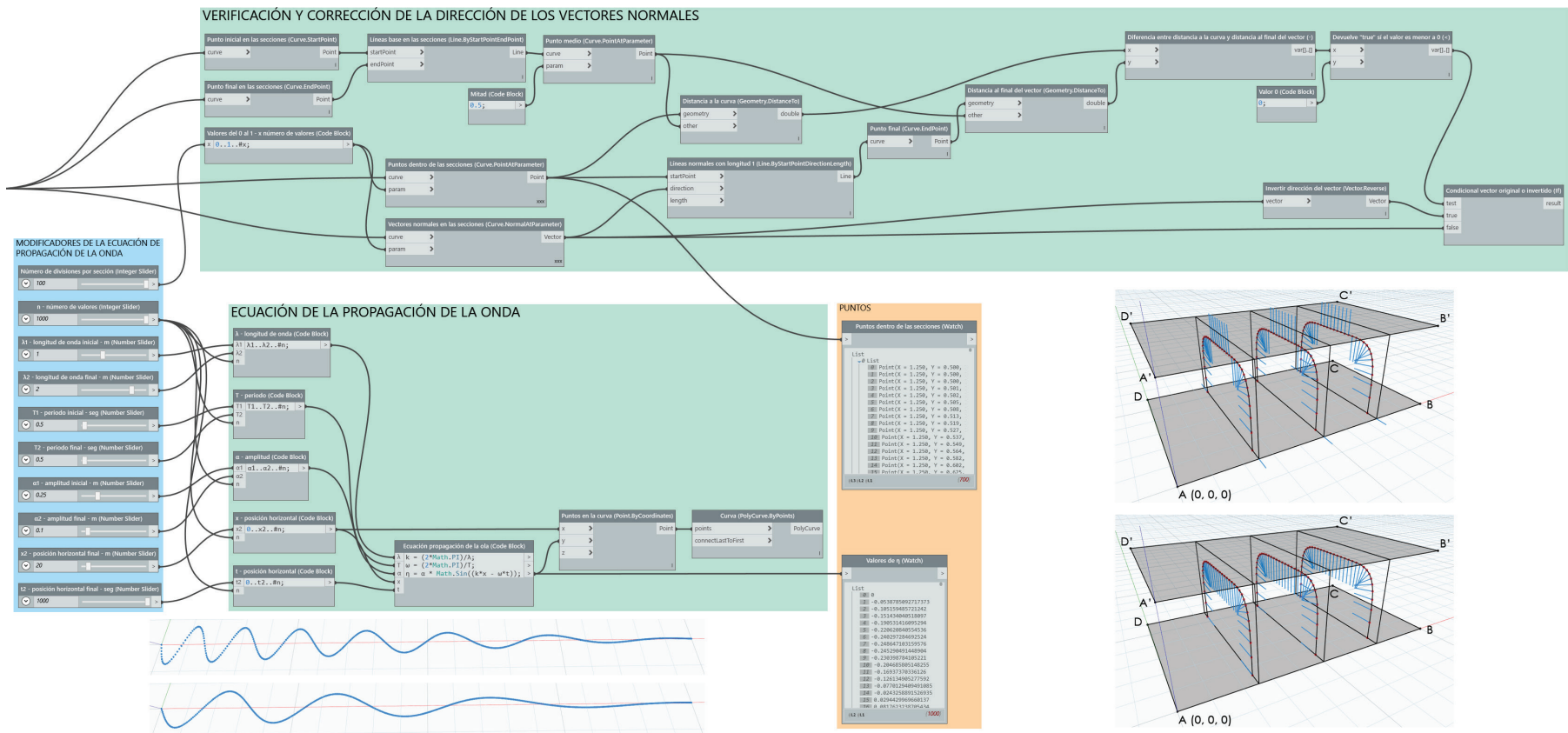


Dependiendo del proceso de interpolación de las curvas las normales pueden tener diferentes sentidos, por lo cual es necesario comprobar el sentido de cada uno de los vectores, y corregirlos de ser necesario, de tal manera que todos los vectores tengan el mismo sentido, es decir que todos apunten hacia el interior del volumen, o que todos apunten hacia el exterior.

Se crea el nodo "Vectores normales en las secciones (Curve.NormalAtParameter)", en la entrada curva se coloca el resultado del nodo "Secciones (NurbsCurve.ByControlPoints)" y en parámetros el resultado del nodo "Valores del 0 al 1 - x número de valores (Code Block)". El resultado es una lista de vectores perpendiculares a la curva en sus puntos

de origen, es necesario comprobar la dirección de cada uno de estos vectores para poder utilizarlos para desplazar los puntos.

Con el fin de comprobar el sentido de los vectores resultado del proceso anterior se crea una lista de líneas auxiliares, que tengan como puntos iniciales el inicio



**Imagen 2.30.** Algoritmo utilizado para comprobar la dirección de los vectores definidos por puntos en las secciones y ecuación de la onda - Captura de pantalla (Elaboración propia, 2016)



de cada curva y como puntos finales el final de cada curva, se extrae el punto medio de cada una de estas curvas para determinar la distancia desde este punto a cada uno de los puntos desde los cuales se crean los vectores, luego se crea una línea con punto de inicio en los puntos donde se crean los vectores, con dirección igual al vector determinado en ese punto y longitud de uno, se determina la distancia desde el punto medio de las líneas auxiliares a el punto de inicio de cada vector y a el punto final de cada línea creada de longitud uno, se resta el primer valor del segundo, si el resultado es negativo, el vector tiene el sentido hacia el exterior del volumen, si el resultado es positivo, el vector tiene sentido hacia el interior del volumen.

Se crean entonces los nodos "Punto inicial en las secciones (Curve.StartPoint)" y "Punto final en las secciones (Curve.EndPoint)", los mismos que tienen como entrada el resultado del nodo "Secciones (NurbsCurve.ByControlPoints)", el resultado de estos nodos se ingresa en las entradas startPoint y endPoint del nodo "Líneas base en las secciones (Line.ByStartPointEndPoint)", el resultado de este nodo son las líneas auxiliares de las cuales se extraerán el punto medio, se conecta a la entrada curva del nodo "Punto medio (Curve.PointAtParameter)", a la entrada parámetros se le conecta el resultado del nodo "Mitad (Code Block)", que contiene el siguiente código:

0.5;

Esto determina el punto medio de las líneas.

Se determina la distancia del punto medio auxiliar a cada punto creado en las curvas mediante el nodo "Distancia a la curva (Geometry.DistanceTo)", este nodo requiere dos geometrías diferentes como entradas y entrega como resultado la distancia comprendida entre las dos geometrías, en la primera entrada se conecta el resultado del nodo "Puntos dentro de las secciones (Curve.PointAtParameter)" y en la segunda entrada se conecta el resultado del nodo "Punto medio (Curve.PointAtParameter)".

El siguiente paso es definir las líneas de longitud uno con dirección igual al vector perteneciente al punto del que inician, se crea entonces el nodo "Líneas normales con longitud 1 (Line.ByStartPointDirectionLength)", se necesita ingresar un valor para el punto de inicio, la dirección y la longitud, en la entrada punto de inicio se utiliza el resultado del nodo "Puntos dentro de las secciones (Curve.PointAtParameter)", para la entrada dirección se ocupa el resultado del nodo "Vectores normales en las secciones (Curve.NormalAtParameter)" y en la entrada longitud no se ingresa ningún valor, el programa por defecto le asigna un valor igual a uno.

El resultado del nodo "Líneas normales con longitud 1 (Line.ByStartPointDirectionLength)" se conecta a la entrada del nodo "Punto final (Curve.EndPoint)" para determinar el punto final de cada una de las líneas que se crearon perpendiculares a las curvas de las secciones.

Luego se determina la distancia desde el punto medio auxiliar y el punto final de las líneas creadas con los vectores con el nodo "Distancia al final del vector (Geometry.DistanceTo)", se conectan a sus entradas los valores de los nodos "Punto final (Curve.EndPoint)" y "Punto medio (Curve.PointAtParameter)".

Para comprobar el sentido de cada vector se resta el valor del resultado del nodo "Distancia a la curva (Geometry.DistanceTo)" del valor del resultado del nodo "Distancia al final del vector (Geometry.DistanceTo)", para esto se crea el nodo "Diferencia entre distancia a la curva y distancia al final del vector (-)". Los valores mayores a cero tienen sentido hacia el exterior del volumen y los valores menores a cero tienen sentido hacia el interior del volumen.

Se utiliza el nodo "Devuelve true si el valor es menor a 0 (<)" para establecer si el valor resultante del nodo "Diferencia entre distancia a la curva y distancia al final del vector (-)" es menor a cero, si el resultado es menor a cero se crea un índice con



valor verdadero, si el resultado es mayor a cero, se crea un índice con el valor falso.

Se tiene entonces una lista con valores verdadero y falso, dependiendo del sentido de cada vector, ahora solo queda determinar que los índices con valor verdadero inviertan el sentido del vector y los que contengan valor falso mantengan su sentido, de esta manera todos los vectores tendrán el mismo sentido.

Para esto se crea el nodo "Condicional vector original o invertido (If)", y se conecta a la entrada test la lista resultado del nodo "Devuelve true si el valor es menor a 0 (<)", además se crea el nodo "Invertir dirección del vector (Vector.Reverse)", a su entrada se conecta el resultado del nodo "Vectores normales en las secciones (Curve.NormalAtParameter)", el resultado es un vector con el sentido invertido al original, este resultado se conecta a la entrada verdadero del nodo "Condicional vector original o invertido (If)", finalmente se conecta a la entrada falso del mismo nodo el resultado del nodo "Vectores normales en las secciones (Curve.NormalAtParameter)". El resultado final de todo el proceso es una lista de vectores con inicio en los puntos definidos en las curvas que conforman las secciones y sentidos iguales.

Nodos resultados en esta sección:

- Puntos dentro de las secciones (Watch)

Nodos modificadores en esta sección:

- Número de divisiones por sección (Integer Slider)"

### 2.3.1.H. ECUACIÓN DE LA PROPAGACIÓN DE LA ONDA

Una vez definidas las secciones, los puntos que se desplazarán y el sentido para valores positivos y negativos, se necesita crear la definición de la curva mediante la aplicación de la ecuación de la propagación de ondas de agua, el resultado de la resolución de esta ecuación determinará la distancia que cada punto se desplaza ya sea en sentido positivo o negativo, una vez aplicado el desplazamiento en todas las secciones, estas se pueden utilizar para generar la superficie.

Se definen los nodos modificadores "n - número de valores (Integer Slider)", con rango de 500 a 1000 e incrementos de 25, " $\lambda_1$  - longitud de onda inicial - m (Number Slider)", con rango de 0.25 a 2.5 e incrementos de 0.25, " $\lambda_2$  - longitud de onda final - m (Number Slider)", con rango de 0.25 a 2.5 e incrementos de 0.25, "T1 - periodo inicial - seg (Number Slider)", con rango de 0.25 a 5 e incrementos de 0.25, "T2 - periodo final - seg (Number Slider)", con rango de 0.25 a 5 e incrementos de 0.25, "a1 - amplitud inicial - m (Number Slider)", con rango de 0 a 1 e incrementos de 0.05, "a2 - amplitud final - m (Number

Slider)", con rango de 0 a 1 e incrementos de 0.05, "x2 - posición horizontal final - m (Number Slider)", con rango de 10 a 100 e incrementos de 5, y "t2 - posición horizontal final - seg (Number Slider)", con rango de 100 a 1000 e incrementos de 20. Estos nodos permiten modificar y definir los valores de longitud de onda, periodo, amplitud y número de puntos que conforman la curva, interpolando entre un grupo de valores iniciales y un grupo de valores finales.

Se crea el nodo " $\lambda$  - longitud de onda (Code Block)", que permite definir un listado de valores para la longitud de onda a partir de un valor inicial, un valor final y el número total de elementos de la lista, se ingresa el siguiente código:

```
 $\lambda_1$ .. $\lambda_2$ ..#n;
```

En donde, la entrada  $\lambda_1$  se conecta a la salida del nodo " $\lambda_1$  - longitud de onda inicial - m (Number Slider)", la entrada  $\lambda_2$  se conecta a la salida del nodo " $\lambda_2$  - longitud de onda final - m (Number Slider)" y la entrada n se conecta a la salida del nodo "n - número de valores (Integer Slider)".

El nodo "T - periodo (Code Block)" define una lista de valores para el periodo de la curva, de igual manera a partir de un valor inicial y un valor final, con un número n de elementos, se utiliza el siguiente código:

```
T1..T2..#n;
```



La entrada T1 se conecta a la salida del nodo "T1 - periodo inicial - seg (Number Slider)", la entrada T2 se conecta a la salida del nodo "T2 - periodo final - seg (Number Slider)" y para definir el número de elementos la entrada n se conecta a la salida del nodo "n - número de valores (Integer Slider)".

Se crea también el nodo "a - amplitud (Code Block)", este nodo define una lista de valores para la amplitud de la curva, con un valor inicial, un valor final y el número de elementos de la lista, se ingresa el siguiente código:

```
a1..a2..#n;
```

la entrada a1 se conecta a la salida del nodo "a1 - amplitud inicial - m (Number Slider)", la entrada a2 se conecta a la salida del nodo "a2 - amplitud final - m (Number Slider)" y la entrada n se conecta a la salida del nodo "n - número de valores (Integer Slider)".

El nodo "x - posición horizontal (Code Block)" determina la longitud total del recorrido de la curva y al mismo tiempo crea una lista para valores de x a partir de un punto inicial, un punto final y el número de valores que conforman la lista, el valor inicial para x se ha preestablecido como 0, se ingresa entonces el siguiente código:

```
0..x2..#n;
```

La entrada x2 se conecta a la salida del nodo "x2 - posición horizontal final - m (Number Slider)" y la entrada n a la salida del nodo "n - número de valores (Integer Slider)".

El nodo "t - posición horizontal (Code Block)" permite establecer el periodo de tiempo en el cual se desarrolla todo el movimiento de las ondas, de igual manera que con x, la variable t inicia desde el valor 0, la lista resultante tiene un valor final para t y el número de elementos determinados por el siguiente código:

```
0..t2..#n;
```

En donde, la entrada t2 se conecta a la salida del nodo "t2 - posición horizontal final - seg (Number Slider)" y la entrada n se conecta con la salida del nodo "n - número de valores (Integer Slider)".

Una vez establecidas las relaciones entre todas las variables que intervienen en la ecuación, se ingresa las operaciones matemáticas que se deben realizar para obtener los valores de  $\eta$  que determinan la elevación para los diferentes valores de x a lo largo del recorrido de la curva.

Con este fin se crea el nodo "Ecuación propagación de la ola (Code Block)", en el cual se definen las operaciones matemáticas mediante el siguiente código:

$$k = (2 * \text{Math.PI}) / \lambda;$$

$$\omega = (2 * \text{Math.PI}) / T;$$

$$\eta = a * \text{Math.Sin}((k * x - \omega * t));$$

Se crean las entradas  $\lambda$ , T, a, x, t. Estas entradas se conectan a las salidas de los nodos " $\lambda$  - longitud de onda (Code Block)", "T - periodo (Code Block)", "a - amplitud (Code Block)", "x - posición horizontal (Code Block)" y "t - posición horizontal (Code Block)" respectivamente.

El resultado final del nodo "Ecuación propagación de la ola (Code Block)" es una lista con los valores de  $\eta$ , la elevación, de la curva, estos valores son los que se utilizarán para desplazar la lista de puntos que conforman las secciones.

Nodos modificadores en esta sección:

- n - número de valores (Integer Slider)
- $\lambda_1$  - longitud de onda inicial - m (Number Slider)
- $\lambda_2$  - longitud de onda final - m (Number Slider)
- T1 - periodo inicial - seg (Number Slider)
- T2 - periodo final - seg (Number Slider)
- a1 - amplitud inicial - m (Number Slider)
- a2 - amplitud final - m (Number Slider)

- x2 - posición horizontal final - m (Number Slider)
- t2 - posición horizontal final - seg (Number Slider)

Nodos resultados en esta sección:

- Valores de  $\eta$  (Watch)

### 2.3.1.1. DEFINICIÓN DE LA SUPERFICIE

Una vez definidas las listas de puntos que conforman las secciones y la lista de vectores según los cuales se desplazarán los puntos, se crean los nodos "Descomponer lista de vectores (Flatten)" y "Descomponer lista de puntos (Flatten)",

que son listas de un solo nivel resultantes de los nodos "Condicional vector original o invertido (If)" y "Puntos dentro de las secciones (Watch)" respectivamente.

Para mover los puntos se crea el nodo "Mover puntos en la dirección del vector según la curva (Geometry.Translate)", este nodo posee las entradas geometría, a la cual se debe conectar la lista de puntos resultante del nodo "Descomponer lista de puntos (Flatten)", dirección, a esta entrada se le debe vincular con la lista de vectores resultante del nodo "Descomponer lista de vectores (Flatten)", y distancia, en esta entrada se ingresa la lista que contiene los valores de  $\eta$  que es resultado del nodo "Valores de  $\eta$  (Watch)".

Ahora se tienen los puntos desplazados según los valores de  $\eta$ , se necesita crear curvas a partir de estos puntos para poder definir las secciones definitivas.

La lista resultante del nodo "Mover puntos en la dirección del vector según la curva (Geometry.Translate)" se conecta a la entrada lista del nodo "Dividir en listas con x elementos (List.Chop)", de esta manera la lista de puntos se pretende dividir en sub-listas que contengan los puntos según cada sección, para esto se conecta la entrada longitud a la salida del nodo "Número de divisiones por sección (Integer Slider)" que define el número de puntos que se creó en cada sección.

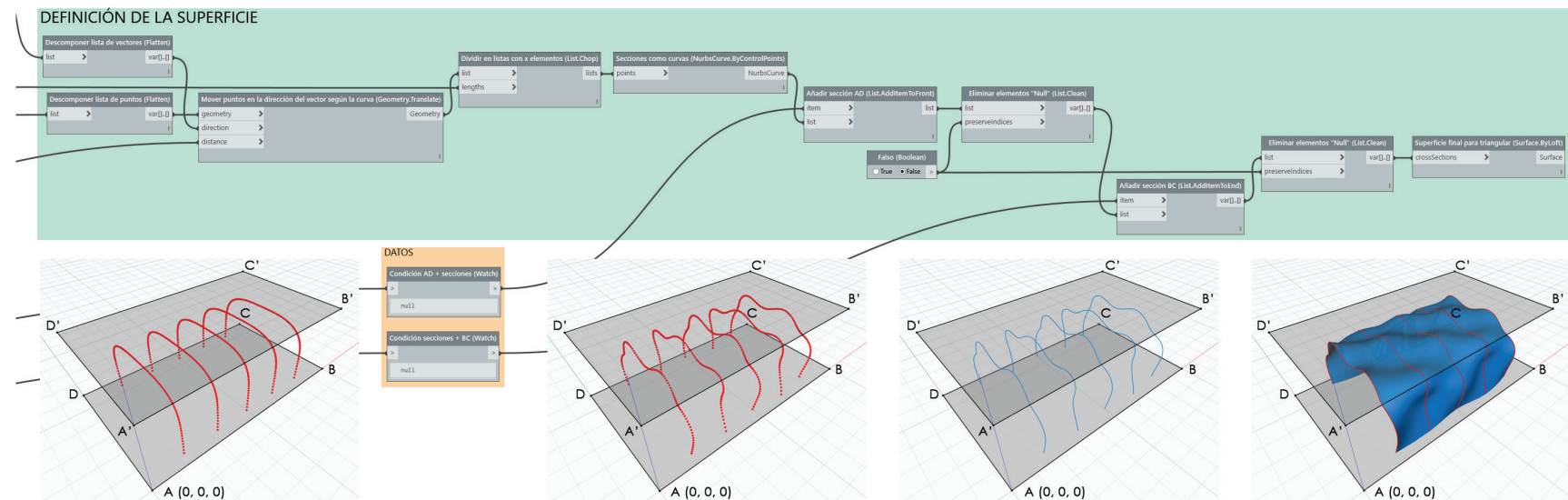


Imagen 2.31. Algoritmo que desplaza los puntos según la curva de las ondas y define la superficie - Captura de pantalla (Elaboración propia, 2016)



Una vez ordenada la lista de puntos en sub-listas por cada sección, se deben crear las curvas, para esto se ocupa el nodo "Secciones como curvas (NurbsCurve.ByControlPoints)", se conecta su entrada puntos a la salida del nodo "Dividir en listas con x elementos (List.Chop)", el resultado es finalmente las secciones definitivas representadas como curvas NURBS.

En este punto se pueden incorporar las laterales AD y BC, en caso de que las variables que determinan si se utilizan o no sean falsas, lo único que se debe hacer es limpiar la lista resultante de los valores null que se hayan generado.

Se crea entonces el nodo "Añadir sección AD (List.AddItemToFront)" para añadir la lateral AD al inicio de la lista, a la entrada lista se conecta el resultado del nodo "Secciones como curvas (NurbsCurve.ByControlPoints)" y a la entrada item se le conecta el resultado del nodo "Condición AD + secciones (Watch)".

Para limpiar la lista en caso de ser necesario se crea el nodo "Eliminar elementos Null (List.Clean)" y se conecta el resultado del nodo "Añadir sección AD (List.AddItemToFront)" a su entrada lista, y el resultado del nodo "Falso (Boolean)" a la entrada preservar índices.

Se realiza algo similar para incorporar la lateral BC, se crea el nodo "Añadir sección BC (List.AddItemToEnd)" que permite que

la lateral BC se incorpore al final de la lista de secciones, a su entrada lista se conecta la salida del nodo "Eliminar elementos Null (List.Clean)" y a la entrada item se conecta el resultado del nodo "Condición secciones + BC (Watch)".

De igual manera en caso de ser necesario se debe limpiar la lista, por lo que se crea un nuevo nodo "Eliminar elementos Null (List.Clean)", en la entrada lista se conecta el resultado del nodo "Añadir sección BC (List.AddItemToEnd)" y a preservar índices el resultado del nodo "Falso (Boolean)".

Finalmente se tiene como resultado el listado completo de secciones que se utilizarán para crear la superficie, como último paso se crea el nodo "Superficie final para triangular (Surface.ByLoft)" que permite crear una superficie a través de la interpolación de las secciones que la conforman, en la entrada secciones cruzadas se ingresa el resultado del nodo "Eliminar elementos Null (List.Clean)", el resultado es la superficie.

A partir de esta superficie se creará la triangulación para la colocación de los módulos.

Nodos resultados en esta sección:

- Condición AD + secciones (Watch)
- Condición secciones + BC (Watch)

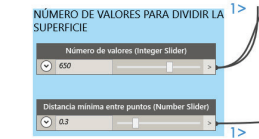
### 2.3.1.J. TRIANGULACIÓN DE DELAUNAY EN LA SUPERFICIE Y EXPORTACIÓN DE ELEMENTOS A REVIT

Una vez que se ha determinado la superficie que contendrá al diseño, se aplica un criterio de parametrización, en este caso una triangulación de Delaunay, para esto es necesario definir un conjunto de puntos a partir de los cuales se creen triángulos de tal manera que las circunferencias que pasan por los vértices de cada triángulo no contengan ningún otro punto del conjunto, esta es la condición que se debe cumplir para la triangulación de Delaunay.

Para generar la lista de puntos ubicados sobre la superficie se utiliza el sistema de coordenadas UV propio de la superficie, se crean dos listas de valores aleatorios que tengan valores de 0 a 1, la primera lista se le asigna a los valores de U y la segunda para los valores de V, el número de elementos que contiene cada lista se debe definir como una variable que se pueda modificar de ser necesario.

Con este fin se crean los nodos "Lista de x valores aleatorios de 0 a 1 en U (Math.RandomList)" y "Lista de x valores aleatorios de 0 a 1 en V (Math.RandomList)", estos nodos generan una lista de valores de manera aleatoria contenidos entre 0 y 1, el número de valores se establece mediante la entrada cantidad de los nodos, se utiliza el resultado del nodo "Número de valores





CAPÍTULO II - DISEÑO DE UN ESPACIO MEDIANTE EL USO DE ALGORITMOS GENERATIVOS  
SEBASTIÁN ANDRÉS RODAS RIVERA - CRISTIAN JHOVANNY BENAVIDES PADILLA 117





(Integer Slider)" para la entrada de los dos nodos U y V.

Se crea el nodo "Generar puntos según coordenadas UV (Surface.PointAtParameter)" para crear los puntos sobre la superficie a partir de las coordenadas de U y V definidas por los valores de las listas anteriores. A la entrada superficie se conecta el resultado del nodo "Superficie final para triangular (Surface.ByLoft)", en la entrada u se conecta el resultado del nodo "Lista de x valores aleatorios de 0 a 1 en U (Math.RandomList)" y en la entrada v se coloca el resultado del nodo "Lista de x valores aleatorios de 0 a 1 en V (Math.RandomList)". El resultado de este proceso es la lista de puntos que se encuentran de manera aleatoria sobre la superficie.

En este paso es necesario realizar un tipo de control sobre los puntos que se han creado, ya que se han generado de manera aleatoria es posible que existan puntos muy cercanos uno de otro como para poder definir una triangulación aceptable, por lo que se define una distancia mínima a la cual se permite que se encuentre otro punto del conjunto, todos los puntos que se encuentren a una distancia menor a la establecida serán borrados y partir de esto se creará una nueva lista de puntos definitiva.

Se crea el nodo "Eliminar puntos muy próximos para computar (Point.

PruneDuplicates)", este tipo de nodo elimina puntos que se consideren duplicados por poseer coordenadas similares a partir de una tolerancia que determina que tan próximos deben estar los puntos para ser eliminados. A la entrada puntos se conecta el resultado del nodo "Generar puntos según coordenadas UV (Surface.PointAtParameter)" y a la entrada tolerancia se conecta el resultado del nodo "Distancia mínima entre puntos (Number Slider)", este nodo tiene un rango de 0.1 a 1 e incrementos de 0.05.

En este paso es necesario realizar un tipo de control sobre los puntos que se han creado, ya que se han generado de manera aleatoria es posible que existan puntos muy cercanos uno de otro como para poder definir una triangulación aceptable, por lo que se define una distancia mínima a la cual se permite que se encuentre otro punto del conjunto, todos los puntos que se encuentren a una distancia menor a la establecida serán borrados y partir de esto se creará una nueva lista de puntos definitiva.

Se crea el nodo "Eliminar puntos muy próximos para computar (Point.PruneDuplicates)", este tipo de nodo elimina puntos que se consideren duplicados por poseer coordenadas similares a partir de una tolerancia que determina que tan próximos deben estar los puntos para ser eliminados. A la entrada puntos se conecta el resultado del nodo

"Generar puntos según coordenadas UV (Surface.PointAtParameter)" y a la entrada tolerancia se conecta el resultado del nodo "Distancia mínima entre puntos (Number Slider)", este nodo tiene un rango de 0.1 a 1 e incrementos de 0.05.

Posteriormente a partir del listado de puntos se definen coordenadas UV para la superficie mediante el nodo "Coordenadas UV sobre la superficie (Surface.UVParameterAtPoint)", a la entrada superficie se conecta la salida del nodo "Superficie final para triangular (Surface.ByLoft)" y a la entrada puntos se conecta el resultado del nodo "Eliminar puntos muy próximos para computar (Point.PruneDuplicates)". El resultado es un conjunto de coordenadas UV para la superficie creada.

Para generar la triangulación de Delaunay se crea el nodo "Triangulación de Delaunay sobre la superficie (Delaunay.ByParametersOnSurface)", el nodo requiere dos entradas, una para los parámetros y otra para definir la cara en la cual se aplicará la triangulación. Se conectan los resultados de los nodos "Coordenadas UV sobre la superficie (Surface.UVParameterAtPoint)" y "Superficie final para triangular (Surface.ByLoft)". como resultado se obtiene un listado de líneas, estas líneas son los lados que conforman cada uno de los triángulos, cuando se encuentran dos triángulos lado con lado, el lado es el mismo, pero en la lista se duplica este valor.



Para poder colocar los módulos que deben conformar la superficie del diseño se requiere definir una lista de puntos, ordenados de tal manera que contenga sub-listas, cada una que contenga los tres vértices de un triángulo.

Para definir los vértices pertenecientes a un mismo triángulo se realiza el siguiente procedimiento.

Se determina el punto inicial y final de cada una de las líneas en la lista resultado del nodo "Triangulación de Delaunay sobre la superficie (Delaunay.ByParametersOnSurface)" con los nodos "Punto de inicio de cada línea (Curve.StartPoint)" y "Punto final de cada línea (Curve.EndPoint)", las listas resultado de la aplicación de estos dos nodos se conectan a las entradas del nodo "Unir listas (List.Join)". El resultado será una lista de puntos de un solo nivel, esta lista contiene al inicio todos los puntos iniciales y al final de la misma los puntos finales.

Se contabiliza el número total de elementos de la lista resultado de "Unir listas (List.Join)" al conectar la salida de este nodo con la entrada del nodo "Contar número de puntos (Count)". La salida de este nodo es el valor total de elementos contenidos en la lista de puntos.

Se conecta el resultado del nodo "Contar número de puntos (Count)" a la entrada n del nodo "Lista de puntos duplicados

(Code Block)", en este nodo se ingresa el siguiente código:

```
{2..(n-1)..6,4..(n-1)..6,5..(n-1)..6};
```

El código le indica al programa tomar los elementos que encuentren en los índices [2], [4] y [5], y a partir de estos índices incrementar en seis cada valor para tomar otros índices hasta que lleguen al índice [n-1], se crean tres sub-listas dentro de la lista general, una para los valores desde [2] hasta [n-1], otra para los valores de [4] hasta [n-1] y otra para los valores de [5] hasta [n-1]. Por ejemplo, en una lista que contiene 20 elementos, con índices del [0] al [19], la primera sub-lista tendría los valores, [2], [8] y [14], la segunda sub-lista tendría los valores, [4], [10] y [16], y la tercera sub-lista tendría los valores, [5], [11] y [17], siendo excluidos de la lista los índices [0], [1], [3], [6], [7], [9], [12], [13], [15], [18] y [19].

Los rangos establecidos anteriormente son resultado de un proceso de prueba y error para verificar cuáles de estos puntos se encuentran duplicados en la triangulación realizada por el programa.

Estos valores determinan los índices que se tienen que eliminar de la lista de puntos para obtener los vértices de cada triángulo sin ningún valor duplicado.

Para determinar la lista a la cual se aplicará este proceso de filtrado, se conecta el

resultado del nodo "Triangulación de Delaunay sobre la superficie (Delaunay.ByParametersOnSurface)" a la entrada del nodo "Contar elementos - líneas (Count)", el resultado de este nodo se conecta a las entradas de los nodos "Número de triángulos (Code Block)" y "Rangos necesarios para unir los puntos de un mismo triángulo (Code Block)".

El código para el nodo "Número de triángulos (Code Block)" es:

```
n/3;
```

Esto divide el número total de elementos para tres, es decir efectivamente devuelve el número de triángulos que comprende la superficie.

El código para el nodo "Rangos necesarios para unir los puntos de un mismo triángulo (Code Block)" es:

```
{0..2,n..(n+2)};
```

El resultado de este nodo es una lista que posee dos sub-listas, la primera contiene los índices [0], [1] y [2], y la segunda contiene los índices [n], [n+1] y [n+2]. Se utiliza para combinar los puntos de un mismo triángulo en una sola lista.

Luego se crea el nodo "Dividir en listas de puntos por cada triángulo (List.Sublists)", este nodo tiene tres entradas, lista, rangos y desfase. A la entrada lista se le conecta

con el resultado del nodo “Unir listas (List. Join)”, a la entrada rangos se conecta el resultado del nodo “Rangos necesarios para unir los puntos de un mismo triángulo (Code Block)” y para la entrada desfase se crea el nodo “Seleccionar rangos cada 3 valores (Code Block)” con el siguiente código:

3;

Y se conecta su resultado a la entrada desfase.

El resultado del nodo “Número de triángulos (Code Block)” se conecta a la entrada cantidad del nodo “Tomar listas según número de triángulos (List.TakeItems)”, para la entrada lista se ocupa el resultado

de la lista “Dividir en listas de puntos por cada triángulo (List.Sublists)”.

El listado resultado del nodo “Tomar listas según número de triángulos (List.TakeItems)” requiere ser aplanado a un solo nivel para poder continuar con el proceso, se realiza mediante el nodo “Descomponer lista (Flatten)”.

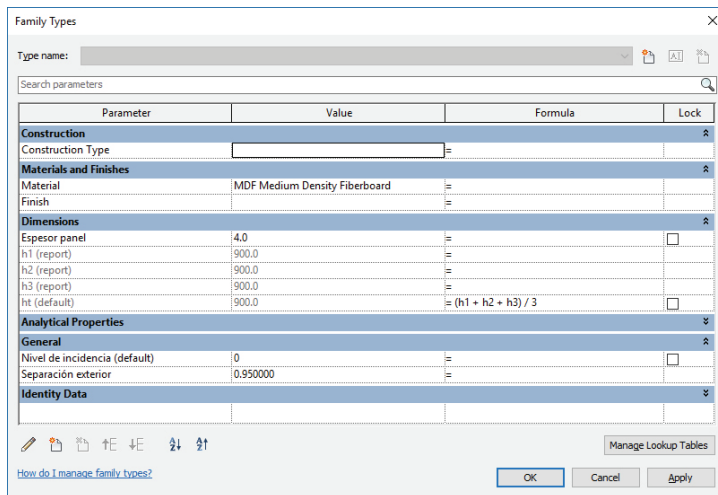
De igual manera se utiliza un nodo “Descomponer lista (Flatten)” para aplanar la lista de índices resultado del nodo “Lista de puntos duplicados (Code Block)”.

Los resultados de los dos nodos “Descomponer lista (Flatten)” se ingresan en el nodo “Eliminar puntos duplicados (List.RemoveItemAtIndex)”, el resultado de

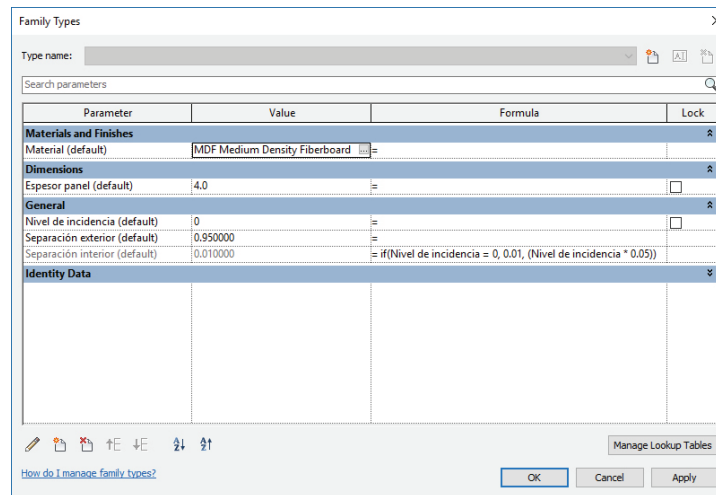
este nodo es una lista de un solo nivel que contiene los vértices de cada triángulo que contiene la superficie, ordenados de tal manera que, al tomar los elementos de tres en tres, se conforman dichos triángulos.

Se invierte el sentido de la lista resultante del nodo “Eliminar puntos duplicados (List.RemoveItemAtIndex)” al conectarla a la entrada del nodo “Invertir lista (List.Reverse)”.

Se necesita dividir la lista resultante del nodo “Invertir lista (List.Reverse)” en sub-listas que contengan tres elementos, para esto se ocupa el nodo “Dividir lista en listas de 3 puntos (List.Chop)”, el resultado es el listado de puntos que se ocuparán para colocar los elementos en Revit.



**Imagen 2.33.** Cuadro Family Types del Módulo A, entorno de trabajo de Revit - Captura de pantalla (Elaboración propia, 2016)



**Imagen 2.34.** Cuadro Family Types del Panel A, entorno de trabajo de Revit - Captura de pantalla (Elaboración propia, 2016)

Además, se crea el nodo "Componente adaptativo (Family Types)" que permite elegir un elemento de entre las varias familias que se encuentran cargadas en el proyecto de Revit que se encuentra actualmente abierto.

### Determinación del Módulo A

El módulo que se utilizará sobre la superficie obtenida en Dynamo, tiene como base el tetraedro abcd, que tiene como centro de masas el punto M, en donde la base está conformada por el triángulo abc, estos puntos se utilizarán para colocar este elemento en el listado de puntos obtenidos mediante el algoritmo definido en Dynamo.

El módulo y sus componentes se creará en el software Revit, gracias a la compatibilidad que tiene con Dynamo, el extraer información directamente desde Revit para procesarla dentro de Dynamo y tras haber sido procesada representarla otra vez en Revit es algo muy sencillo.

Se requiere crear un tipo de familia, las familias dentro de Revit son los elementos que componen el diseño, que permita crear un elemento a partir de tres puntos de inserción. El tipo de familia que permite esto se conoce como componente adaptativo, y coloca cualquier elemento que se haya diseñado en el entorno de creación de familia de Revit, dentro de un proyecto de Revit a partir de una serie de

puntos de inserción, que se conocen como puntos adaptativos.

Dentro de Revit se busca la opción que diga "Familias" y "Nuevo...", esto abre una nueva ventana en la cual están todos los tipos de plantillas de familias que tiene Revit, se busca la opción que diga Metric Curtain Panel Pattern Based.rft y se escoge la opción "Abrir", esta familia está basada en un patrón comprendido por puntos adaptativos, por lo que se le considera como un componente adaptativo.

Al abrirse el entorno de edición de la familia aparece en la ventana una cuadrícula de 3 x 3 cuadrados, el cuadrado interior tiene sus cuatro vértices resaltados, estos son los puntos adaptativos del diseño, pero se requiere que sean solamente tres, para poder cambiar esto se selecciona la cuadrícula y en el panel de propiedades se cambia el tipo de "Rectangle" a "Triangle (Flat)", la cuadrícula cambia inmediatamente de 3 x 3 cuadrados a 3 x 4 cuadrados, en la cual en los dos cuadrados interiores se encuentra un triángulo con los vértices como puntos adaptativos.

Se crean una serie de puntos y líneas de referencia con el fin de poder establecer un tetraedro a partir del triángulo comprendido por los puntos adaptativos, estos puntos serían a, b y c, se toma la distancia perpendicular desde cada uno de los lados del triángulo al vértice opuesto, es decir las alturas, y a cada

una se le asigna un parámetro, para esto se selecciona la cota y en la parte superior derecha se busca la opción "Label" y se hace clic en el botón "Create Parameter", se abre una nueva ventana

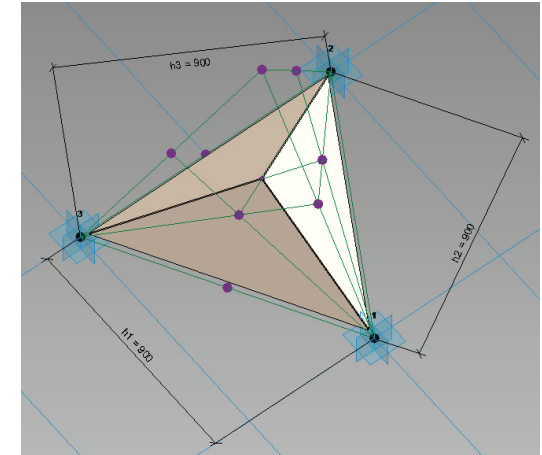


Imagen 2.35. Diseño del Módulo A, entorno de trabajo de Revit - Captura de pantalla (Elaboración propia, 2016)

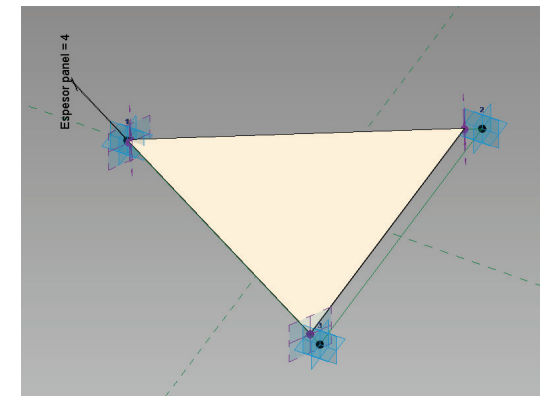


Imagen 2.36. Diseño del Panel A, entorno de trabajo de Revit - Captura de pantalla (Elaboración propia, 2016)



con las propiedades del parámetro, se le establece un nombre y se selecciona la opción "Instance", esto permite que el valor pueda variar entre cada instancia que se encuentre en el proyecto.

Se asignan los nombres h1, h2 y h3 a las alturas del triángulo. Se define al baricentro del triángulo base, y se coloca un punto de referencia, al seleccionar al punto dentro del cuadro de propiedades se busca la opción "Offset" y se hace clic en el recuadro que se encuentra al extremo derecho, se abre una nueva ventana que permite seleccionar un parámetro para este valor, se hace clic en la opción "New Parameter" y se le asigna el nombre ht y se activa la opción "Instance". Dentro del cuadro "Family Types" se ingresa la siguiente ecuación dentro del recuadro Formula del parámetro ht (default):

$$(h1 + h2 + h3) / 3$$

Este define que el valor de desfase del punto definido por el baricentro del triángulo abc, sea igual al promedio de las alturas del mismo triángulo, este punto pasa a ser el punto d del tetraedro.

Se define el centro de masas del tetraedro y se coloca un punto de referencia sobre este, recibe el nombre de punto M. se carga la familia Panel A dentro del módulo y se coloca tres instancias dentro del mismo, la primera utilizando los puntos aMb como puntos de inserción, la segunda con

los puntos bMc y la tercera con los puntos cMa.

Además, se crean los parámetros "Espesor panel", "Nivel de incidencia" y "Separación exterior", solamente para el parámetro "Nivel de incidencia" se selecciona la opción "Instance".

### Determinación del Panel A

Para la creación del Panel A, dentro de Revit se crea una nueva familia y se escoge el tipo Metric Generic Model Adaptive.rft, es similar a la familia de tipo Metric Curtain Panel Pattern Based.rft, pero esta no se basa en una cuadrícula y permite al usuario definir en donde se encuentran y cuantos puntos adaptativos hay.

Se colocan tres puntos de referencia de tal manera que formen un triángulo rectángulo, se seleccionan los puntos y mientras están seleccionados se hace clic en la opción "Make Adaptive", esto convierte la selección de puntos de referencia a puntos adaptativos, utilizando los puntos como vértices se dibuja un triángulo con la herramienta "Reference Line".

Sobre dos de las líneas creadas se colocan dos puntos de referencia en cada una, cuatro en total, uno al inicio y uno al final de cada línea. Se selecciona cada uno de los puntos y en el cuadro de propiedades se cambian los valores "Show Reference

Planes" a "Always" y "Measure From" a "Beginning", a los dos puntos que se encuentran más próximos entre sí y que pertenecen a diferentes líneas se le asigna el valor de "Normalized Curve Parameter" igual al parámetro "Separación interior", a los dos otros puntos se le asigna el parámetro "Separación exterior", se colocan como valores por defecto 0.01 para "Separación interior" y 0.95 para "Separación exterior".

Se crean líneas de referencia entre estos cuatro puntos, el resultado asemeja un trapecio, se seleccionan las líneas de referencia dibujadas y se hace clic en la opción "Create Form" y luego "Solid Form", para crear una extrusión a partir del polígono. Se coloca una dimensión entre el plano inferior y el superior del sólido resultante y se le agrega el parámetro "Espesor de panel", a este se le asigna un valor por defecto de 4 mm.

Se crea también el parámetro "Nivel de incidencia".

En el cuadro "Family Types" de la familia se le asigna al parámetro "Separación interior" la siguiente ecuación dentro el recuadro Formula:

$$\text{if}(\text{Nivel de incidencia} = 0, 0.01, (\text{Nivel de incidencia} * 0.05))$$

Esto establece que, si el valor del parámetro "Nivel de incidencia es igual a 0, entonces





el valor de "Separación interior" será igual a 0.01, mientras que, si el valor de "Nivel de incidencia" es mayor a 0, el valor de "Separación interior" será igual a este valor multiplicado por 0.05.

Todos los parámetros creados en esta familia son de tipo "Instance".

Nodos modificadores en esta sección:

- Número de valores (Integer Slider)
- Distancia mínima entre puntos (Number Slider)

Nodos resultados en esta sección:

- Definición de polígonos (Polygon.ByPoints)
- Comprobación con superficies de cada triángulo (Surface.ByPerimeterPoints)
- Colocar elementos dentro del entorno de trabajo de Revit (AdaptiveComponent.ByPoints)

## 2.4. REPRESENTACIÓN Y EXPLORACIÓN DE RESULTADOS

Cada una de las etapas que se han descrito en los puntos anteriores se combinan en un solo algoritmo, que, a la vez, entrega un único resultado, este es la materialización de elementos geométricos definidos dentro del entorno de trabajo de Revit. No obstante, es posible verificar el resultado de cada uno de los algoritmos que se han creado para definir el elemento final y explorar los diferentes resultados que estos entregan.

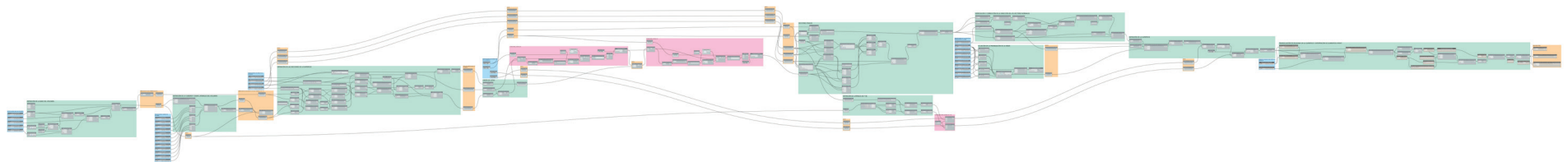
Se puede entonces verificar cada una de las secciones del algoritmo general y modificar los parámetros modificadores para observar los cambios que resultan de esto.

El primer paso es abrir un nuevo proyecto de Revit, y cargar los elementos que se van a utilizar dentro del proceso, es decir, las familias "Módulo A", "Panel A" y "Auxiliar A", además de los materiales "Acrylic" y "MDF Medium Density Fiberboard", que

son los nombres de los materiales que se han utilizado en la definición del algoritmo generativo y posteriormente en los algoritmos del proceso de fabricación.

Dentro de Revit en la sección "Manage" y dentro del grupo "Visual Programming" se hace clic en la opción Dynamo para ejecutar el software como complemento de Revit, desde aquí se abre el archivo con nombre "0. Aplicación del algoritmo generativo.dyn" que contiene todas las definiciones.

El algoritmo finalmente se ejecutará como un solo proceso, es decir como un conjunto de instrucciones para llegar a un fin, por lo que se debe evaluar cada uno de los subprocesos en los cuales sea posible la interacción por parte del usuario, es decir en los llamados grupos de modificadores que se han creado, de esta manera se podrá ver como los cambios en los parámetros definidos a lo largo del algoritmo afectan la totalidad del resultado y la facilidad de generar diferentes opciones de diseño a partir de un solo algoritmo.



**Imagen 2.37.** Representación gráfica del algoritmo generativo, grupos codificados por colores - Captura de pantalla (Elaboración propia, 2016)

### 2.4.1. MODIFICADORES DE LA BASE DEL VOLUMEN

En este grupo se encuentran los nodos "Longitud AB (Number Slider)", "Longitud BC (Number Slider)", "Longitud CD (Number Slider)", "Ángulo ABC (Number Slider)" y "Ángulo BCD (Number Slider)", y permiten la modificación de la superficie que conforma la base del volumen que contendrá al diseño final, el diseño final se adaptará a la forma y tamaño que adopte la superficie de la base.

Se puede generar mediante el ingreso de información en los nodos tanto geometrías regulares como irregulares, los valores para los nodos de longitud aceptan valores de entre 1 a 50, esto en metros, mientras que los nodos referentes a los ángulos aceptan valores entre 5 y 175, este valor es en grados, esto para evitar la posibilidad de que los puntos que definen la superficie creen un polígono que se interseque a sí mismo, sin embargo, ciertas combinaciones

de valores pueden generar un polígono de este tipo, por lo cual es necesaria la interacción del usuario para establecer valores que no generen conflictos en la geometría resultante.

### 2.4.2. MODIFICADORES DE LA CUBIERTA DEL VOLUMEN

En este grupo se encuentra una gran cantidad de modificadores, es en gran parte por el hecho que se ha decidido que cada punto en la cubierta se pueda mover independientemente en las direcciones x, y, z, lo que implica un total de 12 modificadores, el origen de las coordenadas de cada uno de los puntos a partir de los cuales se desplazarán son los puntos de la base.

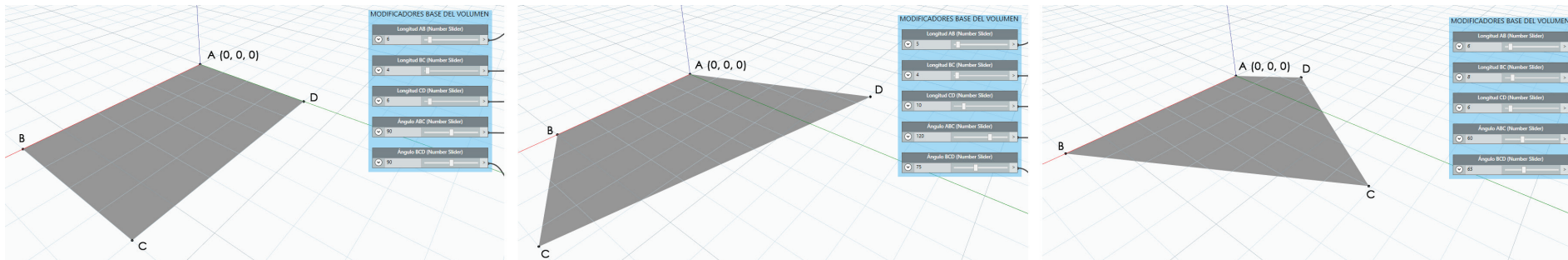
Se tienen entonces los nodos " $\Delta Ax$  (Number Slider)", " $\Delta Bx$  (Number Slider)", " $\Delta Cx$  (Number Slider)", " $\Delta Dx$  (Number Slider)", " $\Delta Ay$  (Number Slider)", " $\Delta By$  (Number Slider)", " $\Delta Cy$  (Number Slider)",

" $\Delta Dy$  (Number Slider)", " $\Delta Az$  (Number Slider)", " $\Delta Bz$  (Number Slider)", " $\Delta Cz$  (Number Slider)" y " $\Delta Dz$  (Number Slider)", todos los modificadores que se refieren a coordenadas en x, y, tienen valores máximos de 10 y mínimos de -10, mientras que los modificadores de las coordenadas z, tienen valores entre 1 y 10.

De igual manera se requiere la interacción directa del usuario para definir valores que no creen conflictos entre las geometrías resultantes, y una vez definida la superficie de la cubierta se puede visualizar el volumen dentro del cual se generará el diseño.

### 2.4.3. MODIFICADORES DE LAS SECCIONES DE LA SUPERFICIE

Estos modificadores permiten determinar el número de secciones que se generará a partir del volumen definido, a lo largo de estas secciones se colocarán puntos de control, que posteriormente se desplazarán



**Imagen 2.38.** Resultados del ingreso de diferentes datos en los parámetros modificadores de la base del volumen - Captura de pantalla (Elaboración propia, 2016)

hacia el interior del volumen, es a partir de estos puntos de control que se interpolará la superficie que albergará al diseño y sus paneles, mientras más puntos se definan el resultado de la interpolación será más definido, siendo lo contrario si se utilizan pocos puntos, sin embargo se debe considerar que si el volumen es un espacio muy reducido colocar gran cantidad de puntos puede generar puntos que se crucen entre los sentidos y, z, lo que dará como resultado un error en la interpolación de la superficie a partir de las secciones, por lo que es vital la interacción del usuario para definir los valores adecuados para computar una superficie sin problemas.

Los modificadores que pertenecen a este grupo están basados en los nodos "Separación - offset (Number Slider)", "Número de puntos en Y (Number Slider)", "Número de puntos en Z (Number Slider)" y

"Número de secciones (Integer Slider)", el primer permite definir la distancia desde las caras del volumen a la cual se colocarán los puntos, el siguiente define el número de puntos que se colocan en la superficie de la cubierta, el tercero define el número de puntos que se colocan en las caras laterales y el último determina el número de secciones que se crearán.

#### 2.4.4. MODIFICADORES PARA LA GENERACIÓN DE LA SUPERFICIE

Este grupo de modificadores básicamente permite seleccionar que caras se considerarán para el cálculo posterior de la superficie, con un total de 4 modificadores, uno para cada cara, AB, BC, CD y AD, y dos opciones diferentes para cada modificador, verdadero o falso, de ser el valor verdadero la cara se computa dentro de la superficie, caso

contrario no lo hace. Existe un total de 16 combinaciones diferentes según los valores de estos parámetros, lo cual generará 16 opciones diferentes para la creación de la superficie.

Los nodos utilizados para definir estos parámetros son "Lateral AB ? (Boolean)", "Lateral CD ? (Boolean)", "Lateral BC ? (Boolean)" y "Lateral AD ? (Boolean)".

La interacción del usuario con este grupo de modificadores es importante para el proceso de creación de la superficie deseada, más no es obligada, puesto que por defecto tienen valores asignados verdadero o falso, esto solo modificará el aspecto de la geometría creada a partir de las secciones, más no afectará la ejecución del algoritmo al punto de poder generar un error.

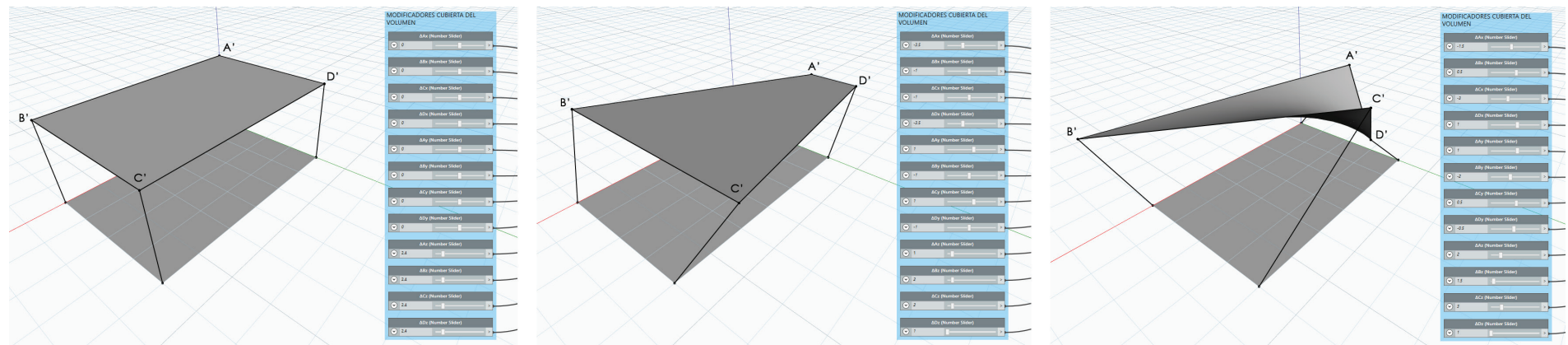


Imagen 2.39. Resultados del ingreso de diferentes datos en los parámetros modificadores de la cubierta del volumen - Captura de pantalla (Elaboración propia, 2016)



### 2.4.5. MODIFICADORES DE LA ECUACIÓN DE LA PROPAGACIÓN DE LA ONDA

En este grupo se encuentran los nodos "Número de divisiones por sección (Integer Slider)", "n - número de valores (Integer Slider)", " $\lambda_1$  - longitud de onda inicial - m (Number Slider)", " $\lambda_2$  - longitud de onda final - m (Number Slider)", "T1 - periodo inicial - seg (Number Slider)", "T2 - periodo final - seg (Number Slider)", "a1 - amplitud inicial - m (Number Slider)", "a2 - amplitud final - m (Number Slider)", "x2 - posición horizontal final - m (Number Slider)" y "t2 - posición horizontal final - seg (Number Slider)". Estos nodos actúan directamente en la ecuación de la propagación de la onda definida anteriormente, para generar como resultado una serie de valores para  $\eta$ , el desplazamiento de los puntos en las secciones.

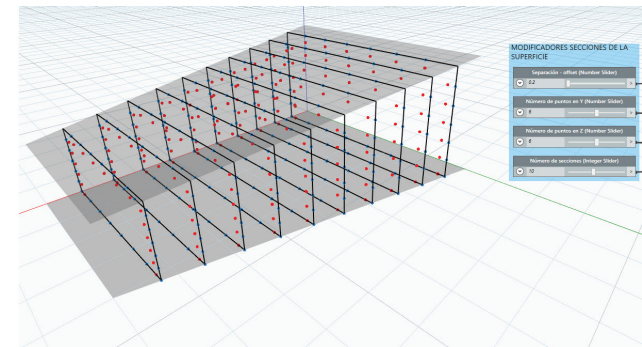
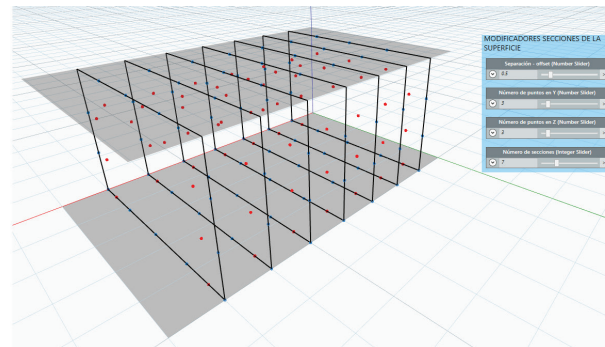
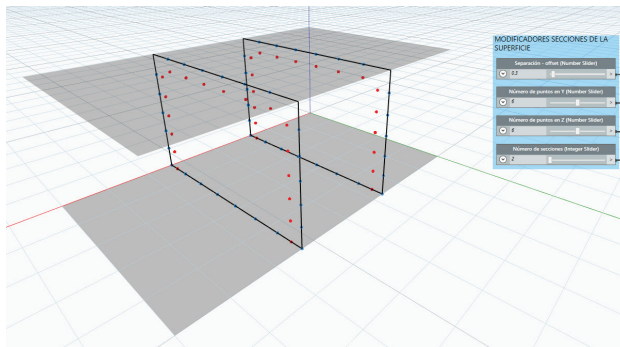
Los nodos pertenecientes a este grupo definen finalmente la superficie al definir

las secciones que se interpolarán para crearla, se tienen modificadores iniciales y finales pues se requiere que la curva generada al inicio sea diferente a la que se genere al final, así los puntos intermedios no son más que una interpolación de estos valores.

El usuario debe explorar los diferentes resultados del manejo de estos parámetros para definir el aspecto más aproximado a lo buscado, se ha podido establecer que el algoritmo funciona muy bien en este punto cuando se trata de volúmenes paralelepípedos, es decir formas regulares, mientras que si la forma del volumen es compleja, se pueden presentar algunos errores en el cómputo de las secciones al trasladar los puntos, es por eso que la interacción del usuario en este punto es fundamental y obligatoria para poder establecer una superficie sin errores.

### 2.4.6. MODIFICADORES PARA LA DIVISIÓN DE LA SUPERFICIE

Este es el último paso en el cual el usuario puede intervenir para determinar la generación del diseño, tras la aplicación de los modificadores anteriores se ha obtenido como resultado una superficie, en la cual se desea ahora colocar los elementos que conformarán finalmente el diseño, para esto se corre un algoritmo para crear una triangulación de Delaunay sobre dicha superficie, para esto se necesita establecer un número de puntos sobre la superficie para conformar la triangulación, el nodo "Número de valores (Integer Slider)" permite definir cuantos puntos se generarán sobre la superficie. El resultado es un conjunto de n número de puntos aleatorios sobre la superficie, al ser aleatorios pueden existir ciertos puntos que se encuentren demasiado cerca como para poder computar una triangulación satisfactoria, por lo que es necesario



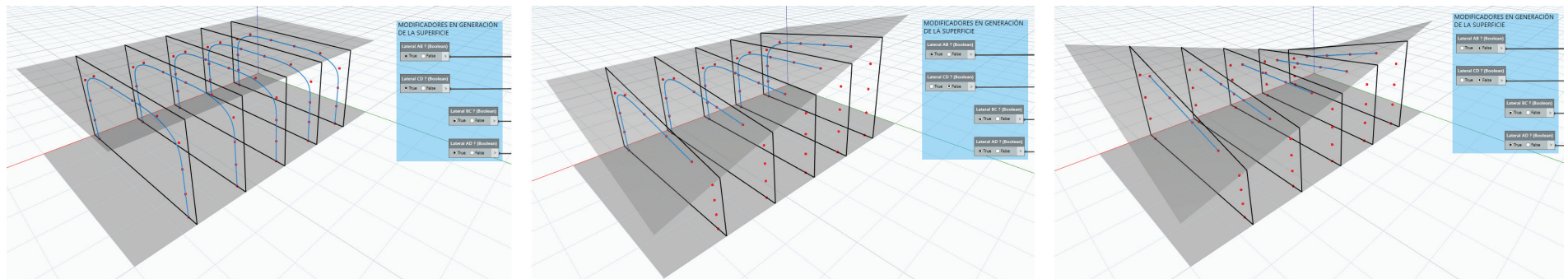
**Imagen 2.40.** Resultados del ingreso de diferentes datos en los parámetros modificadores de las secciones de la superficie - Captura de pantalla (Elaboración propia, 2016)

eliminar los puntos que se encuentren demasiado cerca de otros, para determinar que distancia se considere para eliminar los puntos se utiliza el nodo "Distancia mínima entre puntos (Number Slider)" que establece la distancia a la cual los puntos se consideran como demasiado próximos entre sí y los elimina.

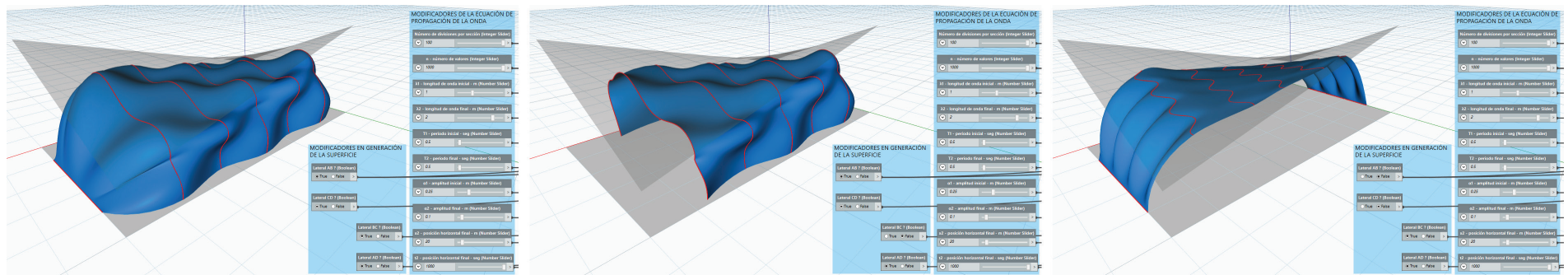
Mientras más grande sea el número de valores para generar los puntos los triángulos resultantes serán más pequeños que si se utilizara más puntos.

Respecto a la distancia entre puntos también permite de cierta manera definir el tamaño mínimo para cada triángulo que se cree en la superficie.

Finalmente se exportan los resultados a Revit, donde se puede apreciar el panelado creado a partir de la triangulación generada como resultado del algoritmo, en donde cada triángulo está compuesto de 3 paneles triangulares.



**Imagen 2.41.** Resultados del ingreso de diferentes datos en los parámetros modificadores de la generación de la superficie - Captura de pantalla (Elaboración propia, 2016)



**Imagen 2.42.** Resultados del ingreso de diferentes datos en los parámetros modificadores de la generación de la superficie y de la ecuación de la onda - Captura de pantalla (Elaboración propia, 2016)



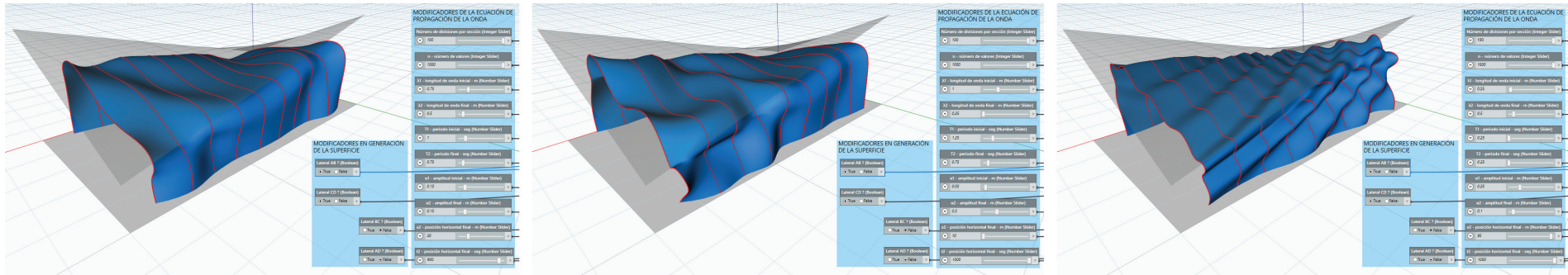


Imagen 2.43. Resultados del ingreso de diferentes datos en los parámetros modificadores de la ecuación de la onda - Captura de pantalla (Elaboración propia, 2016)

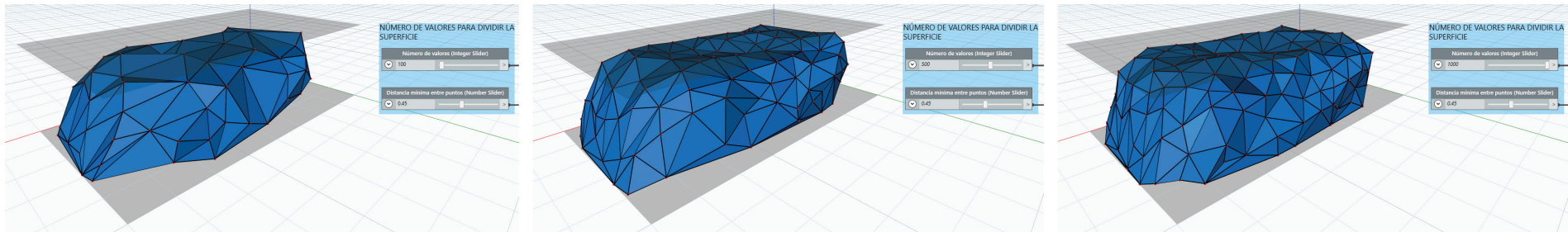
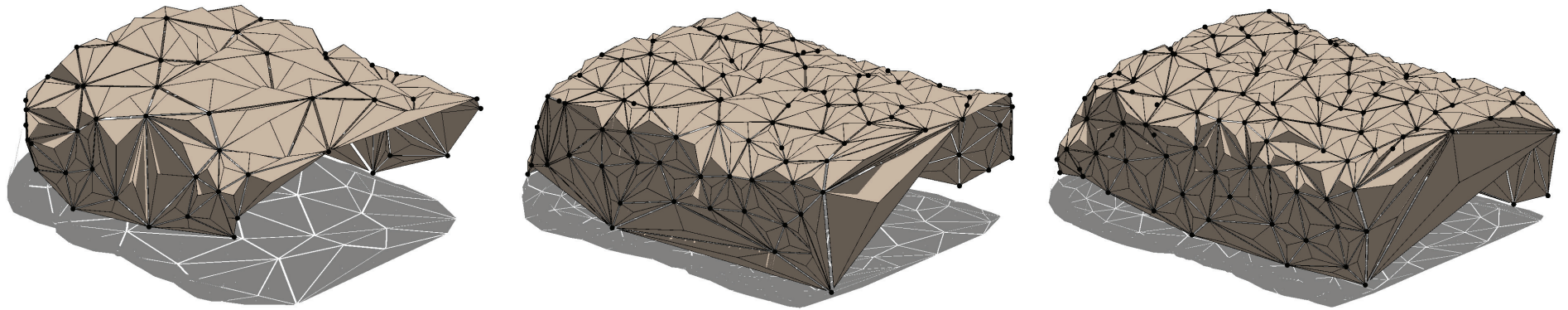


Imagen 2.44. Resultados del ingreso de diferentes datos en los parámetros modificadores para dividir la superficie - Captura de pantalla (Elaboración propia, 2016)



**Imagen 2.45.** Resultados del ingreso de diferentes datos en los parámetros modificadores para dividir la superficie importados a Revit - Captura de pantalla (Elaboración propia, 2016)





## CAPÍTULO III

### 3. ANÁLISIS DEL DISEÑO Y ESTRATEGIAS PARA SU FABRICACIÓN

El diseño final se encuentra compuesto por un número de triángulos casi similar al del número de puntos que se utilizó para la definición de la triangulación de la superficie, en promedio dependiendo del tamaño de los paneles puede ser un número entre 70 y 400 elementos, para un espacio establecido de 24 m<sup>2</sup>, si se tiene en cuenta que cada módulo posee tres paneles, y que entre cada dos módulos continuos se encuentran dos elementos de sujeción, se estaría hablando de entre 500 y 6000 piezas, cada una diferente a la anterior, esto vuelve impráctica la posibilidad de realizar esta tarea de manera manual, se ve la necesidad de utilizar los mismos datos en información que se ha generado en la computadora al ejecutar el algoritmo generativo, para generar un algoritmo que entregue de manera ordenada y codificada un listado de elementos para su posterior fabricación, esto sin mayor intervención del usuario, aparte de hacer clic en ejecutar proceso.

#### 3.1. ESTABLECIMIENTO DEL PROCESO DE FABRICACIÓN ÓPTIMO PARA EL DISEÑO

##### 3.1.1. MAQUINARIA CNC

Se deben entonces considerar varias

maneras en las cuales se pueda llegar a este resultado final, es en este punto donde se hace evidente la ventaja de trabajar con software CAD/CAM, que permite crear diferentes diseños de manera iterativa, para una posterior examinación y exploración del resultado, y que, de manera intuitiva, estos resultados puedan después ser fabricados por maquina especializada CNC, estos reciben la información directamente desde los programas CAD y mediante un programa CAM, generan una serie de instrucciones para que la maquinaria cree las piezas necesarias, establecidas en el diseño.

Maquinaria de Control Numérico Computarizado, se dice de la maquinaria que ha sido adaptada para ser controlada por una computadora, es decir que se ha automatizado, el CNC se puede aplicar a diferentes tipos de maquinarias, dependiendo de los resultados deseados, pueden encontrar tornos, rectificadoras, etc.

La computadora que se incorpora en la maquinaria es la que controla el, posición y velocidad de los motores que accionan los ejes de la máquina, esto ofrece una gran ventaja sobre los sistemas manuales, pues permite crear diseños de gran exactitud en las piezas trabajadas, además de contar con la capacidad de mover varios ejes a la vez para la creación de partes complejas.

Una vez iniciado el proceso de maquinado,

el operador de la maquinaria no requiere modificar ni hacer nada más, esto permite optimizar los tiempos de trabajo.

El término control numérico hacer referencia a los códigos numéricos que se ingresan en la máquina para el proceso de fabricación, estos códigos son una serie de instrucciones que tiene una secuencia lógica.

La maquinaria CNC utiliza el sistema de coordenadas cartesiano para determinar cualquier ubicación en el espacio, es decir cuenta con un eje x, y, z.

Por lo general las máquinas CNC trabajan con dos o tres ejes, pero existe maquinaria más especializada que permite trabajar con cuatro y hasta seis ejes, siendo estos últimos ejes de rotación.

#### 3.1.1.A. TIPOS DE MAQUINARIA CNC

##### Torno CNC

Este tipo de maquinaria permite la mecanización de piezas mediante la revolución de un cabezal donde se sujeta la pieza que se desea trabajar, es así que esta maquinaria permite roscar, cortar, agujerear, cilindrar, desbastar y crear ranuras. Se utilizan herramientas de corte que se mueven contra el elemento que esta sujetado al cabezal, generalmente se utiliza para el mecanizado de materiales duros, como metales.





A los tornos se los considera como el tipo de maquinaria más antigua del mundo, los elementos básicos con los que consta la máquina son el motor y el cabezal principal, la bancada y los carros desplazables, y la UCP o unidad de central de proceso.

### Taladradora CNC

Esta maquinaria realiza un mecanizado en las piezas que tiene por objeto crear perforaciones cilíndricas, para lo cual se utiliza una broca, generalmente se las utiliza industrialmente para el mecanizado de piezas metálicas, de igual manera que los tornos tienen un motor que hace girar a la broca, que al tener contacto con la pieza a maquinar genera las perforaciones.

### Fresadora CNC

En la fresadora se realiza el maquinado de las piezas mediante el arrancado de viruta de material del bloque inicial, para esto se utiliza una herramienta conocida como fresa, esta tiene varios dientes de corte.

Un punto importante a tener en cuenta respecto a este tipo de maquinaria es la versatilidad para maquinar diferentes tipos de materiales, como madera, acero, hierro fundido, diferentes tipos de metales a más de materiales sintéticos.

Son el tipo de maquinaria CNC más polivalente debido a la variedad de mecanizados que se pueden realizar

con ellas. Las fresadoras convencionales mueven el objeto alrededor de la pieza de corte para maquinar la pieza, y las de gama alta y última tecnología permiten mover la pieza de corte en varios ejes.

### Centro de mecanizado CNC

Este tipo de maquinaria es altamente automatizada, gracias a la gran cantidad de herramientas que posee, en algunos casos hasta 100 o más, puede realizar gran variedad de operaciones de mecanizado, todo regido por CNC, la velocidad de este tipo de maquinaria la hace sobresalir de entre el resto, pero esto contrasta con los altos costos que se requieren para el proceso de fabricación. Tienen la capacidad de construir piezas muy complejas con rapidez y exactitud.

### Cortadora láser CNC

Este tipo de maquinado, a diferencia de los anteriores no genera viruta, ya que se realiza al aplicar altas temperaturas a la superficie de un material, el mismo que al calentarse se fusiona y vaporiza, el término láser se refiere a "Light Amplification by Stimulated Emission of Radiation", o "amplificación de luz por emisión estimulada de radiación".

Esta tecnología permite el maquinado de materiales muy duros, se puede mecanizar, por ejemplo, metales duros, metales blandos, aleaciones termo resistentes,

cerámicos, compuestos silicios, cuero, cartón, tejidos, madera, plásticos, entre otros.

El software para el CNC es básico en este tipo de maquinaria, ya que se pueden controlar los diferentes parámetros del haz de luz, y del movimiento del cabezal.

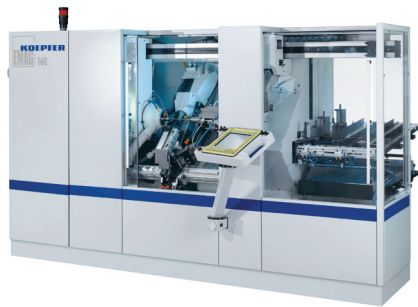


Imagen 3.1. Máquina de torno CNC - Fotografía (CMZ, s.f.)



Imagen 3.2. Máquina de taladro CNC - Fotografía (JPG tools & tools, 2016)





**Imagen 3.3.** Máquina de fresado CNC - Fotografía (DirectINDUSTRY, 2016)



**Imagen 3.4.** Máquina centro de mecanizado CNC - Fotografía (DirectINDUSTRY, 2016)



**Imagen 3.5.** Máquina de cortado láser CNC - Fotografía (DirectINDUSTRY, 2016)

El corte con láser es un tipo de mecanizado rápido y productivo, que permite cortar piezas grandes o pequeñas.

### **Cortadora por chorro de agua CNC**

Esta maquinaria es capaz de cortar un gran rango de materiales, utilizando un chorro de agua a gran presión, o una mezcla de agua y una sustancia abrasiva, generalmente se ocupan abrasivos si se requiere cortar materiales muy duros, como metal o granito. El uso de solamente el chorro de agua a presión se ocupa por lo general para materiales como madera y cauchos.

Se ocupa este método especialmente en casos donde el material que se quiere cortar es altamente sensible al calor y las altas temperaturas que generan otros métodos de corte.



**Imagen 3.6.** Máquina de cortado por chorro de agua CNC - Fotografía (DirectINDUSTRY, 2016)

Es un método muy limpio de corte pues no se genera polvo o gases producto del corte, el agua por lo general se recicla en un sistema de circuito cerrado y en la mayoría de los casos se puede enviar directamente al sistema de alcantarillado local, mediante este método se pueden cortar piezas de hasta 460 mm de espesor, dependiendo de la dureza del material.

Tras haber realizado una comparación entre los diferentes tipos maquinaria CNC y haber visto sus ventajas y desventajas se ha decidido utilizar la máquina de corte láser CNC para trabajar y producir las piezas para el diseño, esto por principalmente por la gran cantidad de materiales que se pueden trabajar mediante este método, el acabado fino que se genera mediante su utilización y los costos relacionados con su operación, además de ser la de más fácil acceso en el medio.

### **3.1.2. ALGORITMOS DE FABRICACIÓN**

Una vez definido el diseño y sus componentes, y haber colocado los elementos sobre la superficie generada, es necesario generar un set o conjunto de instrucciones para que el programa realice el despiece y codificado de los elementos que necesitan ser fabricados para el ensamblaje del modelo.

Se entiende que tras haber ejecutado el algoritmo generativo en el entorno de trabajo de Dynamo, el resultado son todos

los elementos definidos, ya exportados al entorno de trabajo de Revit, los mismos deben ser despiezados, codificados, y ubicados en una vista que permita exportar las geometrías para poder realizar los cortes en la máquina CNC.

Es así que se ha definido una lista de procesos a ser ejecutados para llegar al resultado final, que es los archivos en formato \*.dwg a ser entregados al operador de la maquinaria CNC para producir los cortes de las piezas que conforman el diseño.

Se ha establecido el siguiente árbol de procesos:

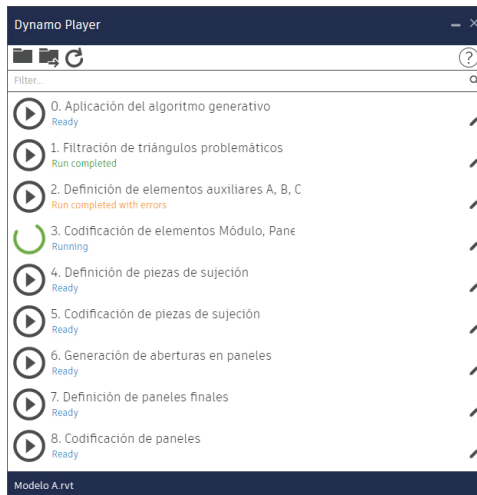
- 0. Aplicación del algoritmo generativo
- 1. Filtración de triángulos problemáticos
- 2. Definición de elementos auxiliares A, B, C
- 3. Codificación de elementos Módulo, Panel y Auxiliares
- 4. Definición de piezas de sujeción
- 5. Codificación de piezas de sujeción
- 6. Generación de aberturas en paneles
- 7. Definición de paneles finales
- 8. Codificación de paneles

A continuación, se explicará lo que realiza cada uno de estos procesos y su creación dentro del entorno de trabajo de Autodesk

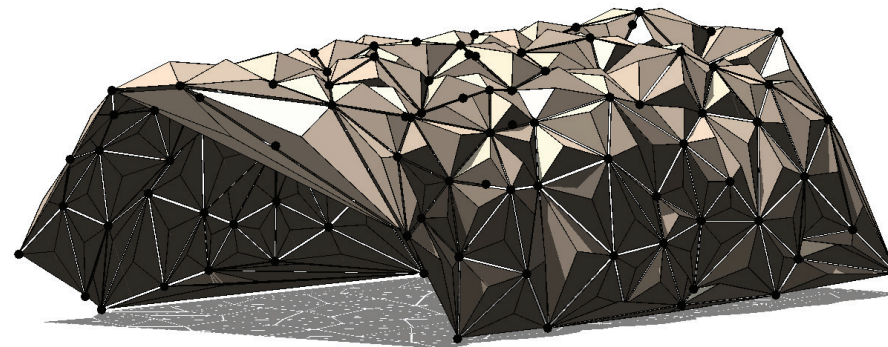
Dynamo como plug-in para Revit, la exploración de los resultados y su posterior representación.

### 3.1.2.A. APLICACIÓN DEL ALGORITMO GENERATIVO

Una vez dentro de Revit, y luego de asegurarse de haber cargado todas las familias y materiales que se utilizan en el proceso de definición del diseño, lo único que falta es ejecutar el algoritmo generativo para incorporar los resultados dentro de Revit, para esto se puede abrir directamente el programa Dynamo y dentro de este abrir el archivo \*.dyn que contiene la definición del algoritmo generativo, o alternatively se puede abrir el plug-in Dynamo Player, este plug-in permite establecer una carpeta en la cual se puede almacenar los diferentes archivos creados en Dynamo para una



**Imagen 3.7.** Ventana de selección de procesos para ejecutar del plug-in Dynamo Player para Revit - Captura de pantalla (Elaboración propia, 2016)



**Imagen 3.8.** Resultado de la ejecución del proceso 0. Aplicación del algoritmo generativo dentro de Revit - Captura de pantalla (Elaboración propia, 2016)

ejecución rápida de los algoritmos dentro de Revit, no se necesita abrir el programa Dynamo en sí, Dynamo Player ejecuta directamente el archivo \*.dyn y todo su código dentro de Revit, una desventaja de este plug-in es la imposibilidad de cambiar los parámetros definidos en los algoritmos, es así que el algoritmo se ejecutará con los valores que se encuentren ingresados en los campos la última vez que se guardó el archivo.

En caso de que se requiera cambiar el valor de alguno de los parámetros es necesario abrir el archivo "0. Aplicación del algoritmo generativo.dyn", y manualmente modificar cada parámetro que sea necesario, caso contrario, directamente desde la ventana del plug-in Dynamo Player se hace clic en el botón reproducir que se encuentra al lado izquierdo del nombre del archivo específico.

Dependiendo de los parámetros definidos dentro del algoritmo, es decir número de

elementos generados y datos computados, puede tardar algunos minutos el trasladar todos los procesos y geometrías a Revit, más aparte de dar clic en el botón ejecutar para correr el proceso, el usuario no necesita modificar nada más en este punto.

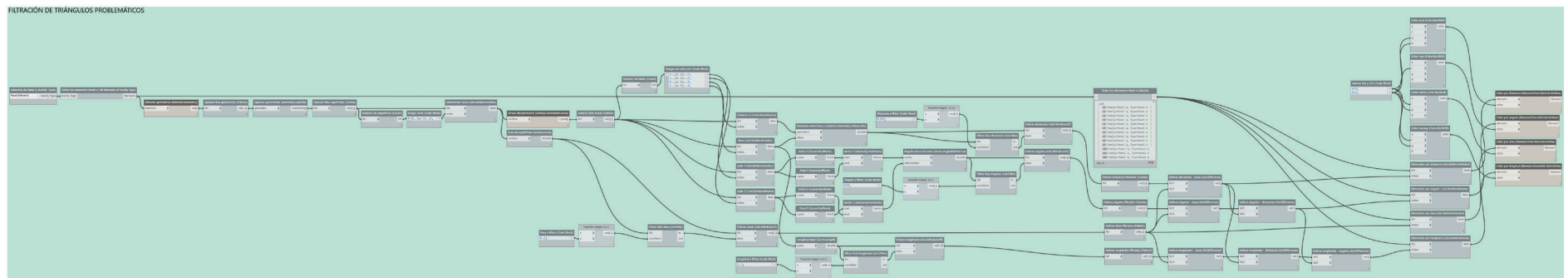
Una vez terminada la ejecución del algoritmo, debajo del nombre del archivo que se acaba de correr aparecerá en texto con letras verdes la frase "Run completed" o ejecución completada en español, en lugar del texto en letras azules "Ready" o listo en español, esto indica que se ejecutó de manera adecuada el algoritmo y no se han encontrado problemas en el proceso, en caso de producirse algún error aparecerá el texto en letras de color naranja "Run completed with errors" o ejecución completada con errores en español, esto puede ser causado por un error en la definición del algoritmo, o por la falta de información que Dynamo extrae de Revit, es decir faltan familias o

componentes que se encuentran en uso dentro del algoritmo.

Al ejecutarse el algoritmo generativo dentro de un archivo de Revit en blanco, al terminar el proceso el resultado serán las piezas o paneles dispuestos sobre la superficie generada.

A partir de este punto no se requiere el ingreso o modificación de los parámetros que componen los siguientes algoritmos, por lo que estos se pueden ejecutar directamente desde el plug-in Dynamo Player.

Aquí se concluye prácticamente el proceso de diseño en sí, los procesos posteriores serán netamente destinados y enfocados a la fabricación del elemento, su despiece y codificación, con la excepción del proceso que genera las aberturas en la superficie, que le permite al usuario interactuar con diferentes resultados al cambiar la posición del sol.

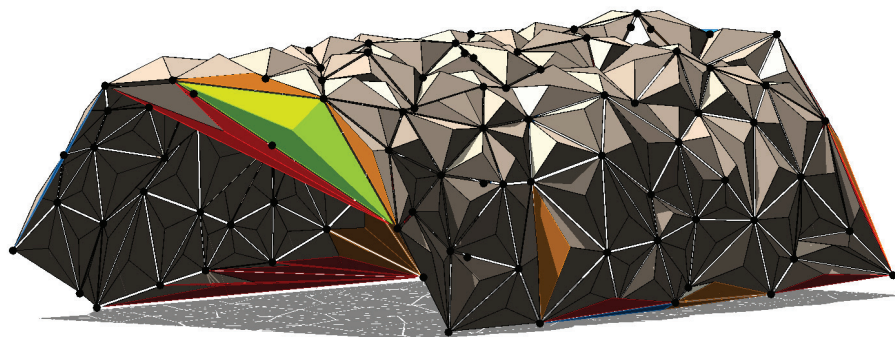


**Imagen 3.9.** Representación visual del algoritmo 1. Filtración de triángulos problemáticos en Dynamo - Captura de pantalla (Elaboración propia, 2016)

### 3.1.2.B. FILTRACIÓN DE TRIÁNGULOS PROBLEMÁTICOS

Una vez se han terminado de importar los datos a Revit para la creación de las geometrías que conforman la superficie del diseño, mediante simple observación de los resultados se puede apreciar que existen ciertas geometrías, paneles, que pueden generar problemas al momento de fabricación del elemento, bien puede ser por su gran tamaño, ángulos muy grandes, lados muy largos, o alturas muy cortas.

Es debido a esto que se ve la necesidad de crear un algoritmo que permita diferenciar de manera objetiva los paneles que cumplen con las condiciones anteriores, y codificarlos de manera visual mediante la aplicación de un color para cada tipo de condición que presente el panel, para así permitir al usuario decidir sobre qué acción tomar en estos elementos.



**Imagen 3.10.** Resultado de la ejecución del proceso 1. Filtración de triángulos problemáticos dentro de Revit - Captura de pantalla (Elaboración propia, 2016)

Se crea entonces el algoritmo "1. Filtración de triángulos problemáticos.dyn" en el que establece el siguiente proceso.

Como primer paso se debe seleccionar mediante un nodo todos los elementos presentes en el archivo de Revit que pertenezcan al tipo de familia "Panel A", de estos elementos se extraen todas las geometrías que los comprenden, el resultado es una lista de sólidos que representan a los paneles.

Los sólidos son luego explotados para poder seleccionar cada una de las caras o superficies que los componen, en total cada sólido contiene 6 caras, se toma una de las dos superficies más grandes de cada sólido y se extrae la información de las curvas que conforman su perímetro, al igual que el área de cada superficie. Cada perímetro está compuesto por un total de 4 líneas.

Se filtra la lista resultante de superficies según los elementos que posean un área mayor a  $0.2 \text{ m}^2$ , por considerarse muy grandes, se ubican los índices que representan estos elementos en la lista de superficies para poder posteriormente extraerlos a una lista diferente.

Con las líneas que componen el perímetro se extraen cada una de ellas en grupos, se determinan los grupos que componen las bases y cubiertas de los polígonos formados, los laterales igual se extraen a una lista aparte.

Se crea un nodo que entregue como resultado la distancia entre las líneas en la lista de base y las líneas en la lista de cubierta, y se filtran todos aquellos valores que sean inferiores a 5 cm, de igual manera se extraen sus índices para poder extraerlos a una lista diferente posteriormente.

Para las listas que contienen las líneas que forman los laterales, se extraen la ubicación de los puntos iniciales y finales de cada una, se define el vector comprendido entre los puntos iniciales y los puntos finales, el resultado será un vector por cada lateral, se comprueba el ángulo comprendido entre estos dos vectores, y se filtran los resultados que sean mayores a  $160^\circ$ , se extraen los índices para extraer estos elementos a otra lista.

Por último, se determina la longitud de las líneas que comprenden las bases, y se



filtran para valores mayores a 1.2 m, de igual manera se extraen los índices para poder tomar estos elementos y moverlos a otra lista.

Se crea una lista a partir de cada una de las operaciones anteriores, en total 4 listas diferentes, pero que pueden contener los mismos elementos, es necesario retirar valores duplicados para poder asignarles a los paneles un color, se establece un rango de prioridad para que elemento se debe mostrar si este cumple dos o más condiciones de las anteriores, para esto se realiza una diferencia entre los valores de las listas de longitud de lado base, ángulos, distancias entre base y cubierta, con la lista de áreas, así la lista de área siempre tendrá prioridad sobre el resto, se realiza la misma operación con las otras listas hasta definir un rango de prioridad, áreas, distancia entre base y cubierta, ángulos, longitud de base, siendo la lista de áreas la de mayor prioridad.

En este punto cada lista contiene un conjunto de valores que representan los

índices de los paneles que se encuentran en Revit, se seleccionan los elementos desde Revit según estas listas, y se les aplica un modificador que represente el elemento en un color determinado.

Los elementos que cumplan la condición de tener un área mayor a 0.2 m<sup>2</sup> se representarán en verde, los que cumplan la condición de tener la distancia entre la base y la cubierta menor a 5 cm se representarán de color azul, si cumplen la condición de tener un ángulo mayor a 160° los elementos se representarán de rojo y finalmente los que cumplan la condición de tener un lado base mayor a 1.2 m se representarán de color naranja.

Una vez corrido este algoritmo, Revit representará todos los triángulos que cumplan con las condiciones anteriores por colores en la vista que se encuentre actualmente activa, ahora se necesita que el usuario realice un análisis rápido para determinar cuáles de los elementos resaltados debe o no ser eliminado, esta parte se realiza de manera manual.

### 3.1.2.c. DEFINICIÓN DE ELEMENTOS AUXILIARES A, B, C

Luego de haber eliminado del diseño aquellos paneles que por ciertas condiciones podrían generar problemas en el proceso de fabricación se necesita definir y colocar una serie de objetos auxiliares, que nos permitan referenciar puntos en el espacio en donde se ubicarán otros elementos y extraer información para la codificación de los paneles.

Se crea de igual un algoritmo aparte en Dynamo para luego ser ejecutado mediante el plug-in Dynamo Player, como primer paso se debe seleccionar todos los elementos "Módulo A" existentes en el proyecto de Revit, cada uno de estos elementos contiene 3 elementos "Panel A", se extrae la información de la ubicación de cada uno de los puntos adaptativos en los que se colocaron los elementos, y se genera una lista de superficies a partir de estos puntos, las superficies corresponden a la triangulación de Delaunay que se generó anteriormente.

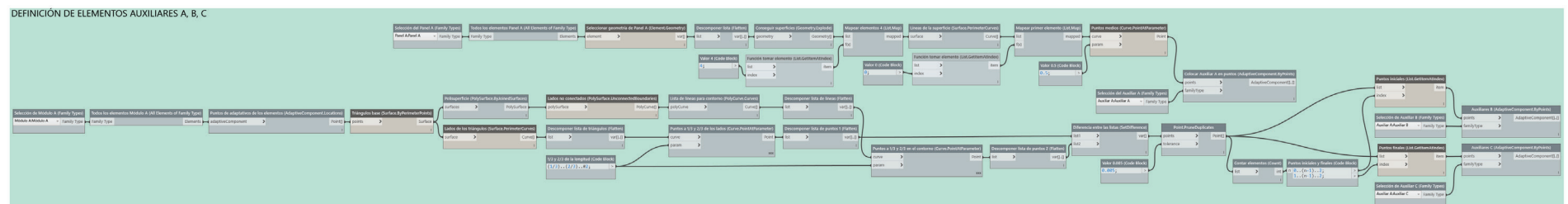


Imagen 3.11. Representación visual del algoritmo 2. Definición de elementos auxiliares A, B, C en Dynamo - Captura de pantalla (Elaboración propia, 2016)



Se crea algoritmo "2. Definición de elementos auxiliares A, B, C.dyn", que buscará los triángulos que definen el exterior de la superficie, puesto que en estos uno de los lados no requiere colocar piezas de sujeción para unirse con otros elementos, para esto se crea una polisuperficie a partir de la lista de superficies creadas, y se extraen aquellas superficies que tienen bordes no conectados y se extraen estos bordes, luego se los descompone en líneas individuales, a estas líneas se les establece un punto al tercio de la longitud total y otro a los dos tercios.

Por otro lado, el algoritmo debe descomponer todos los triángulos resultados de las superficies en las líneas que los definen, de la misma manera se coloca un punto al tercio y otro a los dos tercios de la longitud total de las líneas.

Se hace una operación de sustracción de los elementos de la primera lista a los de

la segunda lista, el resultado será una lista de puntos que se generan solamente en los lados interiores de la polisuperficie, no en el contorno, se eliminan los resultados duplicados y se coloca un elemento "Auxiliar B" en cada punto que se encuentre a un tercio de la distancia total de las líneas y un elemento "Auxiliar C" en los puntos que se encuentren en los dos tercios.

Finalmente se seleccionan todos los elementos "Panel A" que se encuentran en el proyecto, y se extrae la geometría que los compone, es decir los paneles, mediante la aplicación de un nodo se definen los puntos centrales de uno de los lados que componen la base del panel, y en estos puntos se coloca un elemento "Auxiliar A".

Tras ejecutar el algoritmo mediante el plug-in Dynamo Player, se crearán en el proyecto de Revit una serie de elementos

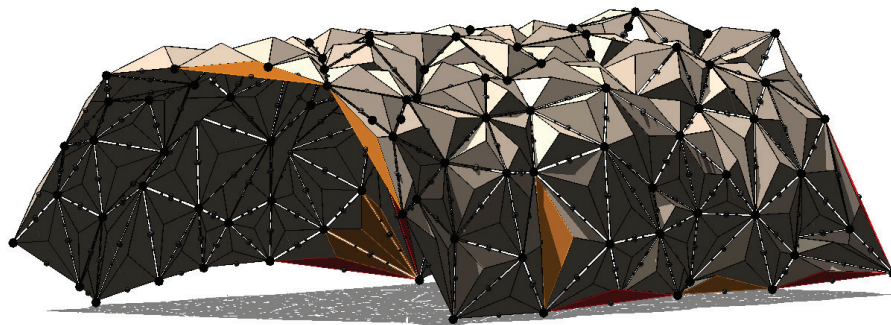
ubicados en los puntos definidos, estos elementos servirán como puntos de referencia para posteriores operaciones que requieren de estos puntos.

### 3.1.2.D. CODIFICACIÓN DE ELEMENTOS MÓDULO, PANEL Y AUXILIARES

Una vez habiendo definido la ubicación de los elementos auxiliares, se asigna una codificación a los elementos "Módulo A", "Panel A", Auxiliar A", "Auxiliar B" y "Auxiliar C", el código sirve para posteriormente en un proceso de fabricación poder unir las piezas sin necesidad de medirlas para descifrar su orden. Para esto se utiliza el parámetro "Mark" que contiene cada elemento de Revit.

Los elementos "Módulo A" tienen el código definido por un valor numérico que empieza en 0 y termina en n-1, siendo n el número total de elementos de este tipo que se encuentran en el proyecto de Revit.

Cada elemento "Módulo A" está compuesto por 3 elementos "Panel A", el código para cada uno de estos elementos comienza con el valor numérico del elemento anfitrión, y continua con un valor literal, a, b, c, para cada uno de los elementos, es decir, para un elemento "Módulo A" de código 27 por ejemplo, existirán tres elementos "Panel A", 27a, 27b y 27c. El orden de enumeración del código es en sentido contrario a las manecillas del reloj.



**Imagen 3.12.** Resultado de la ejecución del proceso 2. Definición de elementos auxiliares A, B, C dentro de Revit - Captura de pantalla (Elaboración propia, 2016)

Los elementos "Auxiliar A" que se han establecido en el paso anterior permitirán definir una relación de proximidad de los elementos "Panel A" entre sí, para de esta manera poder determinar los elementos que se encuentran entre sí. Por tanto, se les asigna el mismo código que a los elementos "Panel A" a los elementos "Auxiliar A", que se encuentran ubicados uno sobre cada elemento "Panel A".

Dentro de Dynamo se define el algoritmo "3. Codificación de elementos Módulo, Panel y Auxiliares.dyn", en donde se seleccionarán todos los elementos "Módulo A" presentes en el proyecto de Revit, se extrae el valor del parámetro "Mark" y de acuerdo al número de elementos se le asigna un valor numérico a cada uno, se reemplaza el valor original dentro de Revit, al valor actualizado por el algoritmo.

Se hace algo similar para los valores de los elementos "Panel A" y "Auxiliar A", primero que nada se tiene que crear una lista que contenga los valores numéricos que se les ha asignado a los elementos "Módulo A" y se concatena este texto con valores literales de a, b y c, esto producirá una lista parecida a la siguiente, 0list { 0list {0a, 1a, ...na}, 1list {0b, 1b, ...nb}, 2list {0c, 1c, ...nc}}, se reordena esta lista en sub-listas que contengan todos los elementos para cada código numérico, por ejemplo, 0list {0list {0a, 0b, 0c}, 1list {1a, 1b, 1c}, ... nlist {na, nb, nc}}, se descompone la lista en un solo nivel y se reemplazan los valores del parámetro "Mark" en los elementos "Panel A" y "Auxiliar A" con los valores de esta lista. Es así que cada uno de los elementos se encuentra ahora codificado según el elemento en el cual se encuentra, es decir al elemento "Panel A".

### 3.1.2.E. DEFINICIÓN DE PIEZAS DE SUJECCIÓN

Esta etapa del proceso de fabricación es una de las más importantes, aquí se definen y crean los elementos que se utilizarán para sujetar un panel con otro, se ha establecido un número total de 2 elementos necesarios para unir y sujetar 2 paneles entre sí, estos se colocarán en los puntos de ubicación de los elementos "Auxiliares B" y "Auxiliares C".

Para unir los elementos de sujeción con los paneles se ha definido un tipo de unión en el cual se genera un corte en los dos elementos, logrando así una traba entre los mismos, esto se realiza en los dos paneles y a los dos lados del elemento que los sujetará.

CODIFICACIÓN DE ELEMENTOS MÓDULO, PANEL Y AUXILIARES

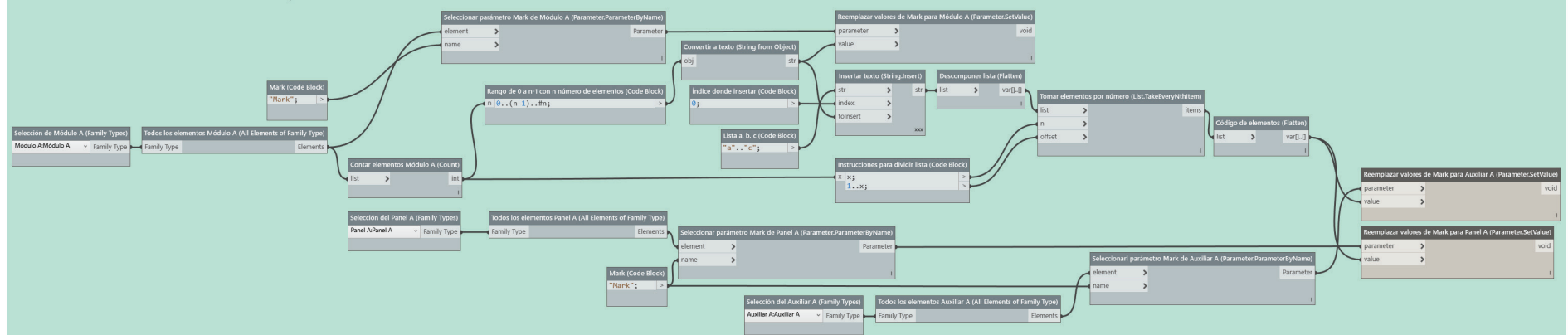
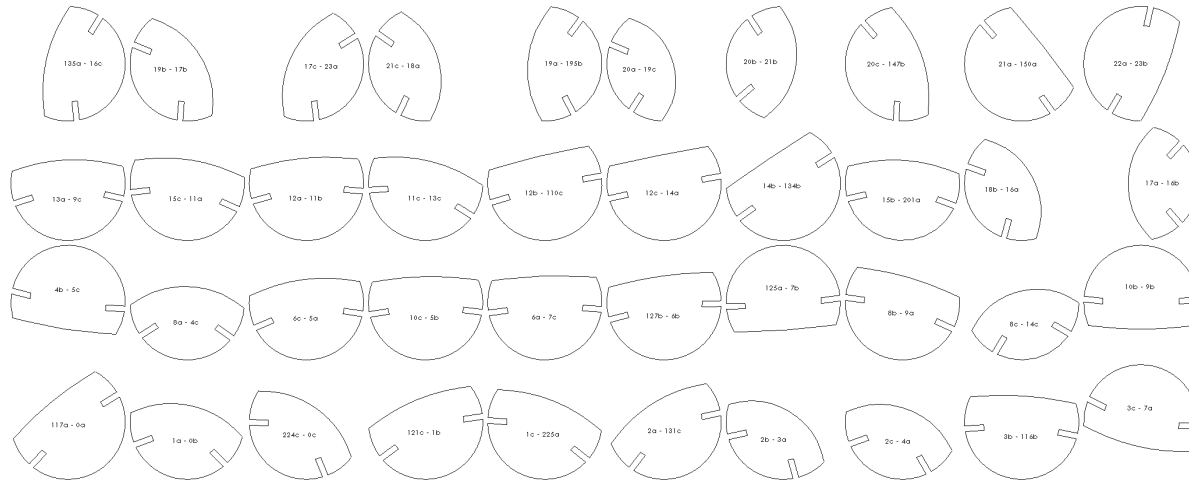


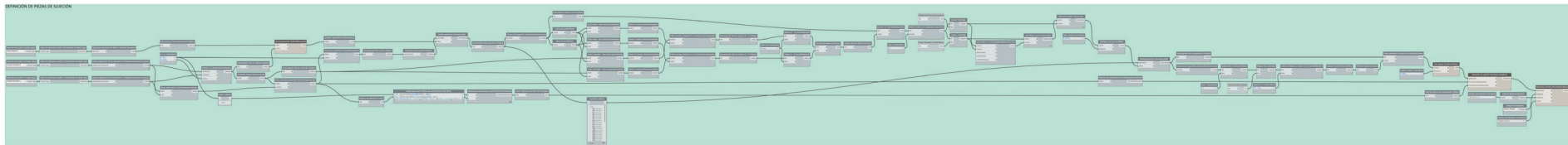
Imagen 3.13. Representación visual del algoritmo 3. Codificación de elementos Módulo, Panel y Auxiliares en Dynamo - Captura de pantalla (Elaboración propia, 2016)



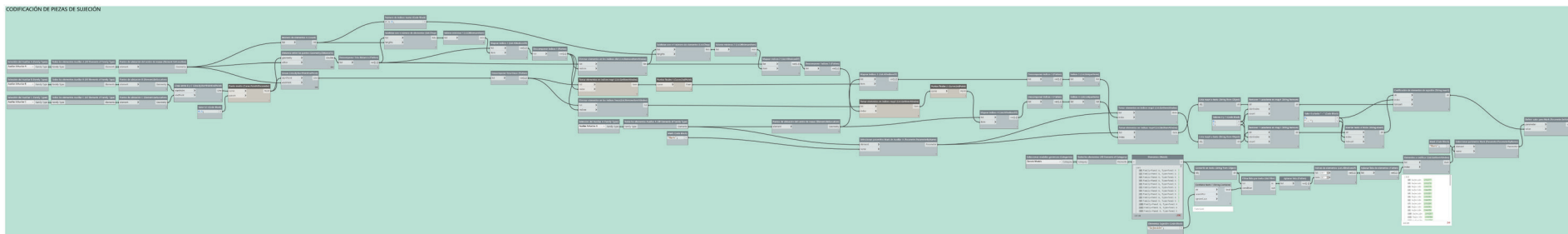
**Imagen 3.14.** Resultado de la ejecución del proceso 4. Definición de piezas de sujeción y 5. Codificación de piezas de sujeción dentro de Revit - Captura de pantalla (Elaboración propia, 2016)

La base para definir los elementos de sujeción es una circunferencia de radio 3 cm, luego de realizarse los cortes para la inserción de los paneles y con el fin de ahorrar material se dividirán las piezas, se tomarán las líneas resultantes de las perforaciones para los paneles y se desfazarán en el sentido contrario a la dirección de los paneles, se unirán estas líneas y se realizará una operación de suavizado del vértice para obtener una curvatura suave, el resultado es una pieza que utiliza menos material y será menos representativa en la superficie.

El primer paso dentro del algoritmo es seleccionar todos los elementos del tipo "Módulo A", "Auxiliar B" y "Auxiliar C", en los elementos auxiliares se extrae la



**Imagen 3.15.** Representación visual del algoritmo 4. Definición de piezas de sujeción en Dynamo - Captura de pantalla (Elaboración propia, 2016)



**Imagen 3.16.** Representación visual del algoritmo 5. Codificación de piezas de sujeción en Dynamo - Captura de pantalla (Elaboración propia, 2016)





Se unen todas las curvas con las líneas, el resultado es una lista de curvas, estas son posteriormente desfasadas 1.5 cm, y se utilizan para dividir las circunferencias creadas al inicio del proceso en dos partes, de las superficies resultado se extraen las curvas que las componen, y se mapea la lista para obtener solamente aquellas superficies que estén compuestas por 12 o más curvas, se extraen los índices de estas curvas para poder seleccionar de

la lista de superficies solamente las que se encuentren en estos índices.

A la lista de superficies se las convierte en sólidos al agregarle un espesor de 3 mm, estas figuras son luego trasladadas a otra ubicación, se utiliza como referencia base el sistema de coordenadas definido por el plano que se colocó en los centros de masas, el nuevo sistema de coordenadas se define con el siguiente código:

$$0..(((r*2)*9)+(0.0025*9))..#10;$$

$$0..(((r*2)*((\text{Math.Ceiling}(n/10))-1))+(0.0025*((\text{Math.Ceiling}(n/10))-1)))..#(\text{Math.Ceiling}(n/10));$$

En donde r es igual al radio de los elementos que se crearon, es decir 3 cm, n es igual al número de puntos creados en los centros de masas. La primera línea de código define los valores de x para una

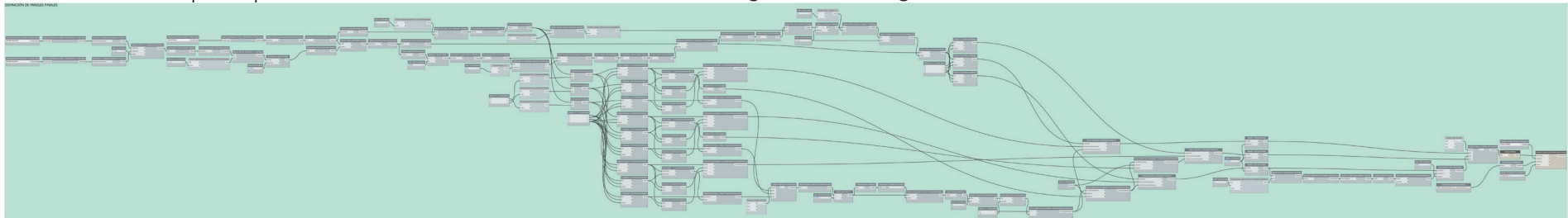


Imagen 3.18. Representación visual del algoritmo 7. Definición de paneles finales en Dynamo - Captura de pantalla (Elaboración propia, 2016)

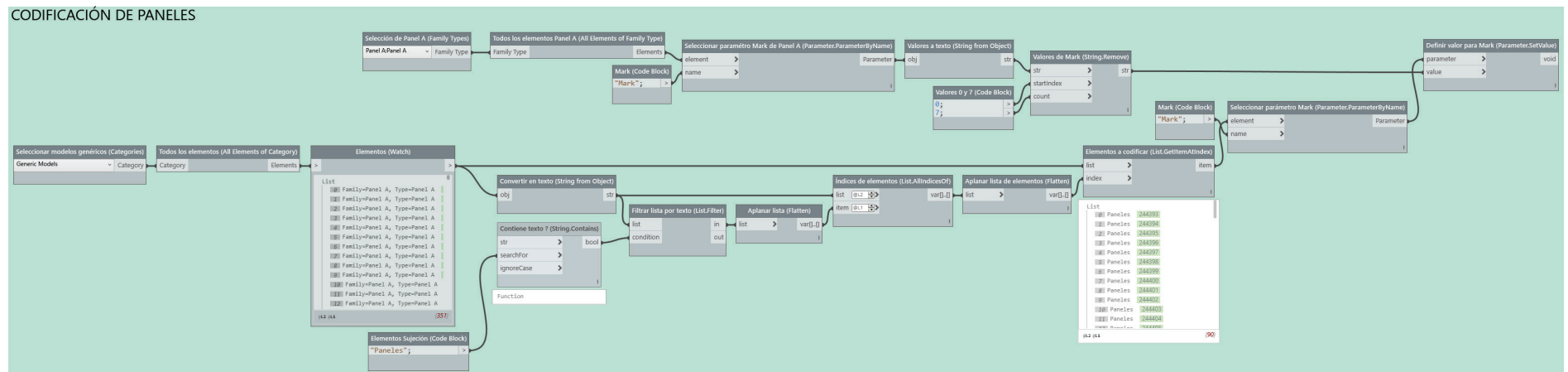


Imagen 3.19. Representación visual del algoritmo 8. Codificación de paneles en Dynamo - Captura de pantalla (Elaboración propia, 2016)



serie nuevos sistemas de coordenadas y la segunda línea los valores de y.

Utilizando los elementos con base en los nuevos sistemas de coordenadas se crean estos elementos como modelos genéricos en Revit, se les agrega las propiedades, material "Acrylic", categoría "Generic Models" y nombre "Sujeción". El resultado es el algoritmo "4. Definición de piezas de sujeción.dyn" cuyo proceso crea elementos geométricos dentro de Revit, a los que se puede modificar las propiedades propias de un elemento de creado en Revit.

### 3.1.2.F. CODIFICACIÓN DE PIEZAS DE SUJECIÓN

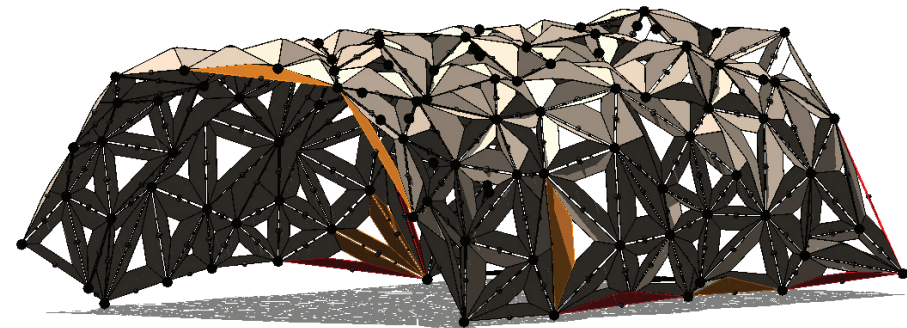
Luego de haber creado los elementos físicos que van a unir y sujetar los paneles entre sí, se necesita codificar estos elementos para hacer posible el proceso de ensamblaje, dado el gran número de piezas resultantes de estos procesos, esto solamente puede ser factible mediante la aplicación de algoritmos que automaticen esta tarea, ya que hacerlo manualmente sería una tarea muy compleja y tediosa para ser realizada efectivamente.

Con este motivo se crea el algoritmo "5. Codificación de piezas de sujeción.dyn", el algoritmo en cuestión toma todos los elementos auxiliares, "Auxiliar A", "Auxiliar B" y "Auxiliar C" y extrae la ubicación de cada uno de estos elementos en el proyecto de Revit.

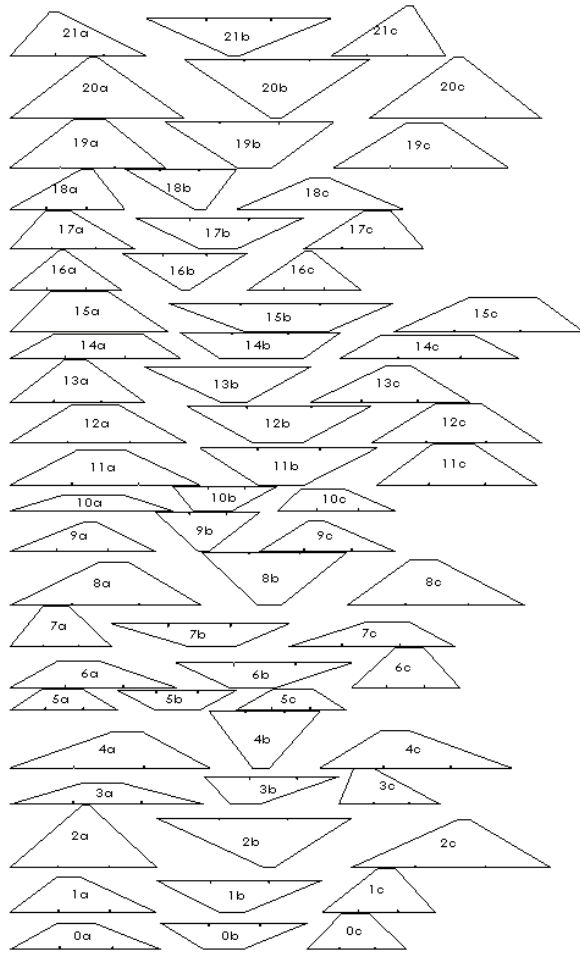
Se crea una línea entre los puntos de B y C, y se define el punto medio de esta línea, luego se crea una serie de líneas con punto inicial en el punto medio entre B y C y punto final en todos los puntos de A, se extrae la longitud de cada uno de estas líneas, luego se debe dividir la lista de distancias en sub-listas con longitud igual al número de elementos A, los dos valores más bajos en cada sub-lista representa los paneles más cercanos al punto que se encuentra en ese índice, de esta manera se puede establecer cuales con los paneles que se encuentran próximos a los elementos de sujeción, se mapea la lista de distancias para devolver el valor mínimo y se extraen estos índices, luego se quita estos índices de la lista de distancias y se vuelve a aplicar el mismo proceso, de esta manera los resultados serán los segundos valores más bajos, de igual manera se extraen los índices de estos elementos.

Se ocupan los índices extraídos para eliminar elementos de la lista de líneas, y se definen los puntos finales de estas líneas, y se utilizan para mapear su ubicación dentro de la lista de puntos de A, y se obtienen los índices de los dos puntos más cercanos para cada elemento A, estos índices se ocupan para extraer la información contenida en el parámetro "Mark" de los elementos "Auxiliar A", se combinan los valores del parámetro "Mark" del punto más cercano con el valor del segundo más cercano en un solo valor, y esto se reemplaza dentro de los valores de "Mark" para los elementos que contengan en su nombre "Sujeción", es decir las piezas definidas en el proceso anterior.

El resultado final le da a cada pieza de sujeción un valor para "Mark" que contiene el código del panel más cercano unido con el código del segundo panel más cercano, necesario para el ensamblaje.



**Imagen 3.20.** Resultado de la ejecución del proceso 6. Generación de aberturas en paneles dentro de Revit - Captura de pantalla (Elaboración propia, 2016)



**Imagen 3.21.** Resultado de la ejecución del proceso 7. Definición de paneles finales y 8. Codificación de paneles dentro de Revit - Captura de pantalla (Elaboración propia, 2016)

### 3.1.2.G. GENERACIÓN DE ABERTURAS EN PANELES

El siguiente paso es definir el valor del parámetro "Nivel de incidencia" de los elementos "Módulo A", este parámetro define el nivel de abertura que se generará en los paneles que conforman el elemento en relación a la perpendicularidad de la cara con el vector proveniente de la ubicación del sol, se han establecido valores de 0 a 10, este paso corresponde más a un criterio de diseño y de experimentación con las capacidades del programa, que funcional, puesto que los elementos diseñados están previstos a ser ubicados en un espacio interior.

Con este fin se ha creado el algoritmo "6. Generación de aberturas en paneles.dyn", lo primero que se hace es seleccionar todos los elementos del tipo "Módulo A", y se extrae su ubicación dentro del proyecto de Revit, a partir de estos puntos se crean una lista de polígonos y una lista de superficies, de la lista de polígonos se extrae el baricentro, mientras que de la lista de superficies se extrae el vector normal a la superficie en el punto donde se ubica el baricentro.

Para comprobar el nivel de perpendicularidad de cada una de las caras con relación al vector comprendido por la incidencia del sol, se extrae dicho vector, esto es posible gracias a la compatibilidad de Dynamo para trabajar

nativamente con Revit, y se realiza el producto del mismo con el vector normal a la superficie ubicado en el baricentro de la misma, el resultado será una lista con valores de 0 a 1, siendo 0 una superficie paralela al vector de la incidencia solar y 1 una superficie completamente perpendicular.

Una vez definida esta lista de valores, se la mapea tomando el valor más bajo y reemplazándolo por 0, y el valor más alto y se lo reemplaza por 10, luego se redondean los valores al entero más cercano, se tiene entonces como resultado una lista de valores de 0 a 1 que contiene solamente valores enteros.

Se selecciona el parámetro "Nivel de incidencia" de los elementos "Módulo A" y se reemplaza el valor actual por el valor conseguido en la lista, Dynamo sobre escribe estos parámetros para cada elemento "Módulo A" dentro de Revit, al ser un parámetro que afecta la geometría del elemento, este se actualiza automáticamente para representar la abertura en los paneles.

### 3.1.2.H. DEFINICIÓN DE PANELES FINALES

Los paneles tal como se importaron desde Dynamo al ejecutar el primer algoritmo necesitan ser modificados antes de estar listos para su fabricación, de la misma manera que se creó unas perforaciones en los elementos para la sujeción de los



paneles, se crea una perforación en los paneles mismos, estas perforaciones se encuentran en los puntos en los que se colocaron los elementos "Auxiliar B" y Auxiliar C", de esta manera se creará una traba entre los paneles y los elementos de sujeción que impedirá que estos se deslicen, el algoritmo tiene el nombre "7. Definición de paneles finales.dyn".

Se obtiene la ubicación de todos los elementos "Auxiliar B" y "Auxiliar C", luego se crea un cilindro entre estos puntos con un radio de 2 cm, se explota la geometría en superficies y se seleccionan las dos caras circulares, a estas se les agrega un espesor igual a 3 mm.

Por otro lado, se seleccionan los elementos "Panel A" dentro del proyecto y se extrae la geometría definida por estos, es decir los paneles en sí, y se realiza una operación de sustracción, a los paneles se les sustrae los sólidos definidos por los cilindros de 3 mm de espesor, esto genera las perforaciones en las que se anclarán las piezas sujetadoras.

El algoritmo además crea un sistema de coordenadas en un punto en la base de cada panel, a partir del cual se recolocarán los paneles en el plano xy, para luego ser importados nuevamente en Revit como modelos genéricos.

Para esto, una vez definidos los paneles con las perforaciones se explota la geometría

para poder extraer las superficies que la componen, se busca seleccionar las caras de mayor área del panel, por lo que se filtra la mapea la lista de áreas para devolver como resultado solamente las superficies mayores a 100 cm<sup>2</sup>, estas son las caras principales de los paneles, estas superficies se descomponen en las curvas de su perímetro, y se extraen los puntos iniciales de estas, y se mapea con los puntos que comprenden las superficies originales de los paneles, en la cara inferior, es decir entrega solamente los puntos que coinciden con los puntos de la cara inferior de los paneles antes de realizar cualquier operación, finalmente se divide la lista en tres sub-listas, cada una conteniendo los puntos los paneles a, b, c, de cada "Módulo A".

Para crear los sistemas de coordenadas de cada triángulo se utilizan los puntos iniciales de las curvas que conforman la cara inferior del panel original, se utiliza un punto como origen, otro para definir el eje x, y otro para definir el eje y de los sistemas de coordenadas, se crea un grupo de sistemas por cada cara del "Módulo A", es decir para a, b, c.

Además, se crea un nuevo sistema de coordenadas hacia donde se moverán los paneles para alinearse al plano xy, para esto se inicia en el origen y se incrementa en x el largo de la base de los triángulos a, luego se incrementa el largo de la base de los triángulos b, para el eje y se realiza

algo similar, pero teniendo en cuenta la altura de los triángulos, y el número total de elementos "Módulo A". el resultado es una lista de sistemas de coordenadas con 3 elementos para x, y n número de elementos en y, siendo n el número de elementos "Módulo A". De esta manera los elementos no se sobrepondrán cuando se importen en Revit.

Con el fin de conseguir una configuración aproximada a la óptima en la disposición de elementos dentro de Revit tras ser importados se ha establecido que los triángulos pertenecientes al grupo b, deben ser rotados 90° para de cierta manera formar una especie de franjas horizontales que desperdiciarán menor cantidad de material y requerirán de menor intervención del usuario para la organización de los mismos antes de mandarlos al centro de corte para su producción.

Cada uno de los grupos de paneles, a, b, c, se les agrega espesor igual a 3 mm, después se toman los paneles b, y se crea un cuadro delimitador con base en los sólidos que los comprenden, se convierte el cuadro delimitador en un sólido con el fin de poder determinar su centro de masas, luego se crea un plano utilizando el centro de masas como origen y como normal el eje z predeterminado, en base a esto, se rotan los paneles b, utilizando estos planos como referencia en 180°.



Una vez realizado este ajuste se unen las tres listas en una sola y se importa como un todo a Revit, con categoría "Generic Models", material "MDF Medium Density Fiberboard" y con nombre "Paneles".

Al ejecutar el algoritmo desde el plug-in Dynamo Player, el resultado serán los paneles colocados en el nivel 1, solamente falta codificar estos elementos para poder ser fabricados.

### 3.1.2.1. CODIFICACIÓN DE PANELES

El último paso en la serie de instrucciones necesarias para poder importar finalmente los resultados de todos los procesos al programa Revit para posteriormente generar los archivos para usarse en las máquinas CNC para crear las piezas.

El algoritmo que se creó para este fin tiene el nombre de "8. Codificación de paneles.dyn", y lo que hace es seleccionar todos los elementos de la categoría "Generic Models" y filtra los resultados que contengan el texto "Paneles" en su nombre, se extraen los índices que corresponden a estos elementos y se selecciona solamente los "Paneles".

Como ya se definió anteriormente la codificación para los elementos "Panel A", ahora se va a seleccionar todos los elementos de este tipo y extraer la lista de valores del parámetro "Mark", donde se encuentra el código de cada

panel, utilizando los valores de esta lista se reemplazan los valores actuales del parámetro "Mark" en los elementos "Paneles".

Utilizando la herramienta "Tag" dentro de Revit se crea un texto sobre cada elemento que contenga el código definido mediante todos los procesos, estos resultados son posteriormente exportados a un archivo de formato \*.dwg para enviar a la máquina CNC.

## 3.2. ANÁLISIS Y ELECCIÓN DE MATERIALES

El siguiente análisis se realizará sobre los materiales que son adecuados para ser cortados por láser, que es el método de producción definido en el trabajo, se comparará diferentes aspectos de cada uno, tanto referentes al material propiamente, como a la acción del láser sobre el mismo.

El corte láser generalmente se utiliza para aplicaciones industriales, para cortar los diferentes materiales el haz de luz que se crea se enfoca en a través de lentes, el material puede reaccionar de varias maneras al ser expuesto por el haz de luz, se puede derretir, quemar, vaporizarse o es soplado por un chorro de gas, el último deja un acabado muy fino en la superficie cortada.

Algunos materiales no se pueden cortar con métodos de corte diferente a este.

Existen tres tipos principales de láser, el láser CO2 que se utiliza para corte, perforado y grabado, el láser de Neodimio (Nd) y el láser de Neodimio Itrio-Aluminio-Granate (Nd-YAG), estos dos últimos tipos se utilizan tanto para grabado como para perforado principalmente, pero se utilizan en aplicaciones que demandan una cantidad de energía alta a muy alta. Los láseres de CO2 son los que se destinan principalmente para el corte de varios materiales, entre ellos acero suave, aluminio, acero inoxidable, titanio, papel, cera, plásticos y tejidos.

### 3.2.1. TIPOS DE CORTE Y LOS MATERIALES

Hay varios métodos de corte, que se utilizan para diferentes materiales, los más utilizados son vaporización, fundir y soplar, fundir y quemar, estrés térmico, grabado, corte en frío, y corte láser estabilizado por combustión.

#### 3.2.1.A. CORTE LÁSER POR VAPORIZACIÓN

El haz de luz enfocado sobre la superficie calienta el material hasta su punto de ebullición, esto produce una perforación, gracias a esta perforación se produce un aumento repentino de absorbencia del material, lo cual hace más profundo a la perforación de manera muy rápida. Mientras el material se encuentra hirviendo, los vapores generados corroen las paredes, ensanchando así la perforación.



Los materiales que se utilizan comúnmente con este método son madera, carbón, plásticos termo formados, etc.

### 3.2.1.B. CORTE LÁSER POR FUNDIDO Y SOPLADO

Este proceso se conoce también como corte por fusión, y utiliza gas altamente presurizado para soplar el material fundido del área de cortado, no requiere tanto poder como otros métodos y evita tener que subir la temperatura del material hasta el punto de ebullición. Este método se utiliza generalmente para cortar metales.

### 3.2.1.C. CORTE LÁSER POR ESTRÉS TÉRMICO

Los materiales frágiles son particularmente sensibles a la fractura térmica, se aprovecha esta propiedad en este proceso de corte por láser. Se enfoca un haz de luz sobre la superficie, causando gran cantidad de

calor localizado, esto a la vez produce una expansión térmica. El resultado, una grieta que puede ser guiada mediante el movimiento del haz de luz. Usualmente se utiliza este método para cortar vidrio.

### 3.2.1.D. CORTE LÁSER REACTIVO

También conocido como corte láser estabilizado por combustión, es similar al corte con soplete de oxígeno, pero con un haz de luz como la fuente de ignición, es más utilizado para cortar acero al carbono de espesores de más de un 1 mm, mediante este método se pueden cortar placas de acero de gran espesor sin la necesidad de mucho poder (Wikipedia, 2016).

### 3.2.2. MATERIALES

Los tipos de materiales y espesores que se pueden cortar varían según la potencia del láser, pero por lo general se clasifican en las siguientes categorías:

- Plásticos
- Laminas metálicas
- Espumas
- Otros

### 3.2.2.A. PLÁSTICOS

Dentro de este grupo se consideran a todos los materiales que consisten de un rango de compuestos orgánicos sintéticos o semi-sintéticos, por lo general son polímeros orgánicos de alta masa molecular, la mayoría son hechos de derivados del petróleo. La plasticidad de estos materiales que les permite deformarse de manera permanente sin romperse es una de las cualidades de esta categoría.

Como ventajas sobre otros materiales, su bajo costo, versatilidad e impermeabilidad al agua.



**Imagen 3.22.** Grabado láser sobre ABS o Acrilonitrilo Butadieno Estireno - Fotografía (Cutlasercut, 2015)



**Imagen 3.23.** Grabado láser sobre PMMA o acrílico - Fotografía (Cutlasercut, 2015)



**Imagen 3.24.** Grabado láser sobre PP o Polipropileno - Fotografía (Cutlasercut, 2015)



### ABS - Acrilonitrilo Butadieno Estireno

Este tipo de material se puede cortar muy bien como láminas, presenta ciertos problemas cuando se incrementa el espesor del elemento cortado, pero puede cortarse hasta los 6 mm de espesor, hay que tener cuidado al cortarlo pues emite gas de cianuro, como material propiamente dicho se dobla mucho antes de romperse.

### PMMA - Acrílico o Polimetilmetacrilato

Este material se corta extremadamente bien, tras el corte se aprecia un borde casi pulido, se puede cortar en espesores de hasta 12 mm, no se debe tener en cuenta ninguna consideración especial al cortarlo, es muy seguro, este material presenta una resistencia al rayado y a la intemperie superior a la de otros plásticos, tiene una alta resistencia al impacto, casi 20 veces superior a la del vidrio y es muy ligero.

### PP - Polipropileno

Se utilizan láminas de este material de hasta 6 mm, es un material flexible y resistente, que permite un trabajado muy fácil, las caras de este material tienen texturas diferentes, la una es muy llana y suave, mientras que la otra presenta cierto texturado, es parcialmente cristalino, es muy liviano y posee gran resistencia, posee el grado más alto de resistencia a los impactos de todos los elementos termoplásticos, pero al entrar en contacto con el láser suele tender a fundirse.

### Poliestireno

Este material tiene gran resistencia al impacto y es muy flexible, pero posee un punto de fusión muy bajo, lo que representa ciertas desventajas a su utilización con corte láser, se puede cortar hasta en espesores de 2 mm.

### 3.2.2.B. LÁMINAS METÁLICAS

Esta categoría hace referencia a láminas o planchas metálicas de poco espesor, tienen un costo elevado comparado con otros materiales que se presentan en este análisis y presentan mayores dificultades al momento de trabajarse mediante láser.

Son materiales muy resistentes, pero en contraparte mucho más pesados que el resto de materiales.

Principalmente se utiliza la tecnología láser para grabar planchas metálicas, más el corte es posible mediante la utilización de láseres de altísima potencia, esto por ende incrementa los costos de fabricación de estos materiales.

### Acero inoxidable

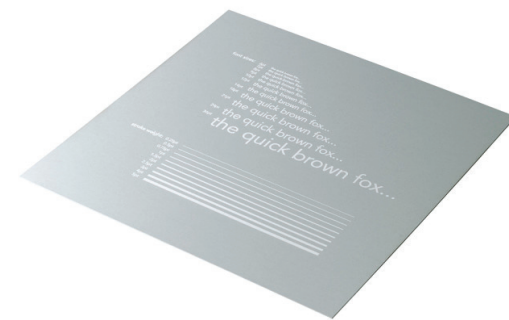
Este material es en realidad una aleación de acero con cromo, contenidos del



**Imagen 3.25.** Grabado láser sobre Poliestireno - Fotografía (CutlaserCut, 2015)



**Imagen 3.26.** Grabado láser sobre acero inoxidable - Fotografía (CutlaserCut, 2015)



**Imagen 3.27.** Grabado láser sobre aluminio - Fotografía (CutlaserCut, 2015)

10% al 12%, presenta gran resistencia a la corrosión, además de las propiedades propias del acero, para poder cortar este material con tecnología láser se requiere de una máquina de gran potencia, pudiendo realizarse cortes en láminas de hasta 1.5 mm, principalmente se utiliza para grabarlo.

### Acero suave

Este es un tipo de acero que posee un porcentaje muy bajo de carbono en su composición, es resistente, pero no realmente templado, es el tipo de acero más utilizado debido a su bajo costo, son más fáciles de manejar y trabajar, para corte láser se requiere una maquinaria de gran poder para poder cortarlo, y se puede cortar en espesores de hasta 1.5 mm, la tecnología láser se usa de mayor manera para grabar este material que para cortarlo.



**Imagen 3.28.** Grabado láser sobre espuma de Polietileno - Fotografía (Cutlasercut, 2015)

### Aluminio

Este material es un metal suave, no magnético, y dúctil, tiene baja densidad y es resistente a la corrosión, al igual que los elementos anteriores, no puede ser cortado por máquinas láser de CO2 y requieren máquinas más potentes, principalmente se utiliza para grabar su superficie.

### 3.2.2.C. ESPUMA

Las espumas son materiales que se forman al atrapar burbujas de gas o líquido en su interior, el volumen de gas es generalmente alto en la mayoría de las espumas, existen dos tipos principales de espumas, de célula cerrada y de célula abierta, en los primeros el gas se encuentra en espacios pequeños y está rodeado por completo por material sólido, en el segundo caso las burbujas de gas se conectan una con otra, las espumas responden de muy buena manera al corte con láser.



**Imagen 3.29.** Grabado láser sobre tejido de lino - Fotografía (Cutlasercut, 2015)

### XPS – Espuma de Poliestireno Extruido

Este tipo de material es utilizado generalmente para construir modelos a gran escala, y modelos voladores a control remoto, debido a su peso extremadamente bajo, este material puede ser cortado muy limpiamente por el láser, además este material no absorbe agua ni se humedece, se puede cortar en espesores de hasta 6 mm.

### Espuma de Polietileno

Este tipo de espuma es especialmente dura, son un tipo de espuma de célula cerrada de mediana densidad, posee una gran capacidad de recuperación de su forma original después de un impacto por lo que se lo suele utilizar para embalaje, se puede cortar en espesores de hasta 15 mm, se comporta muy bien frente a la acción del corte láser.



**Imagen 3.30.** Grabado láser sobre cartón blanco - Fotografía (Cutlasercut, 2015)

### 3.2.2.D. OTROS

En este grupo se incluyen todo tipo de material que no se haya englobado en los principales grupos anteriores y que sea factible cortar con láser.

#### Tejidos

La mayoría de los tejidos reaccionan muy bien a la acción del láser, los sintéticos inclusive reciben una especie de sellado al fundirse el material, lo que impide que este de deshilache, algunos de los diferentes tipos de telas que se pueden cortar son, algodón, fieltro, seda, lino, encaje, poliéster, tejido polar, tela jean, neopreno.

#### Papeles

Los papeles y cartones son elementos ideales para realizar prototipos, la facilidad con la que el láser es capaz de cortar estos materiales les agrega una ventaja

comparativa frente a otros materiales, sobre todo si lo que se requiere es velocidad, este tipo de materiales tienen una composición química similar a la de las maderas, esto permite que el papel se evapore al paso del láser, este se escapa en forma de humo, este aleja el calor del material, de esta manera se evita terminar con bordes quemados, los diferentes materiales que se pueden cortar de este tipo incluyen cartulina, carton, cartón corrugado, papel de hasta 600 gramos.

#### Gomas

Las gomas como la mayoría de los elastómeros se cortan bien con láser, sin embargo, los bordes pueden tener marcas de quemado, por lo general se utilizan espesores hasta de 6 mm, algunos de los materiales más comunes son goma para láser, goma natural, caucho sintético, caucho de silicona, gomaespuma.

#### Maderas

Diferentes tipos de madera se pueden cortar mediante láser, principalmente se corta MDF, balsa, plywood y madera maciza de diferente procedencia, dependiendo del tipo de madera que se utilice el resultado puede ser diferente al ser sometida al láser.

La madera sólida se corta muy bien, sin embargo, hay que tener cuidado en especies de madera que tienen alto contenido de aceite, pues se pueden prender en fuego al entrar en contacto con el láser.

Las maderas compuestas, como el plywood, pueden no tener los mismos resultados que la madera maciza, puesto que estas tienen un alto contenido de pegas, lo que puede de cierta manera afectar el proceso de corte.



**Imagen 3.31.** Grabado láser sobre caucho de silicona - Fotografía (CutlaserCut, 2015)



**Imagen 3.32.** Grabado láser sobre plywood - Fotografía (CutlaserCut, 2015)



**Imagen 3.33.** Grabado láser sobre MDF - Fotografía (CutlaserCut, 2015)



El MDF es una buena elección para cortar, pero tiene una desventaja, y es que experimenta un grado mayor de carbonización en los bordes del corte.

Cada uno de los diferentes tipos de madera poseen propiedades específicas en cuanto a su resistencia, pero por lo general son una buena opción por ser livianas, generalmente permite cortes en espesores de hasta 12 mm.

### 3.2.3. MATERIALES SELECCIONADOS

Luego de hacer una comparación entre los materiales expuestos anteriormente se ha establecido que se utilizarán MDF y PMMA (acrílico) como materiales del diseño.

Los paneles necesitan ser lo más livianos posible, por lo que se ha decidido utilizar MDF, en específico con un espesor de 3 mm, al estar compuestos de madera, los paneles responderán muy bien a la aplicación de pegamentos de carpintería.

Para las piezas de sujeción entre paneles se requiere un elemento que sea transparente, pero a la vez muy resistente, es por eso que se ha pensado en el PMMA (acrílico), ya que este es uno de los plásticos que mejor reacciona a la acción del láser, es muy liviano, tiene gran resistencia y es muy seguro para cortar, se ha establecido un espesor similar al de los paneles, es decir 3 mm, se requiere pegar las piezas de acrílico a los paneles de MDF.

### 3.3. ELABORACIÓN DE UN PROTOTIPO A ESCALA

Como último paso de este trabajo, se ha decidido poner en obra todo lo creado mediante la aplicación de algoritmos para probar la factibilidad que posee el diseño, y de esta manera también el diseño paramétrico, de ser construido en base a los resultados entregados enteramente por la computadora al procesar la información proporcionada por el usuario.

Parece de gran importancia poder representar los resultados de esta investigación de manera real, por lo que se ha decidido crear una maqueta a escala real, con los materiales propuestos en el punto anterior, sin embargo, si bien el tamaño de las piezas es en escala real, la maqueta no será de tamaño completo, pues se tomará una muestra, compuesta de un número pequeño de elementos de un diseño más grande. Esto principalmente por la gran cantidad de elementos que serían necesarios maquinar para la creación de un diseño en el cual se pueda ingresar, y los costos de esto sería muy elevados.

Una muestra, es decir una pequeña parte de un todo, puede demostrar cómo funcionan todas las piezas del conjunto, al igual que permite apreciar la intencionalidad del diseño en términos estéticos.

#### 3.3.1. DEFINICIÓN DE LA MUESTRA

Primero se necesita correr el algoritmo, para este caso específico se ha ingresado una superficie de 8 x 5 metros como base, y una altura de 3 metros a la cubierta, el resto de parámetros se han dejado sin modificar de los predeterminados en el algoritmo generativo.

El resultado contiene un total de 226 elementos "Módulo A" y 678 elementos "Panel A", del total de la superficie del diseño se toma una muestra de 15 elementos "Módulo A", que representan alrededor de 3 m<sup>2</sup> de área proyectada, el resto de elementos son eliminados del proyecto.

Tras haberse definido los elementos que se utilizarán para el modelo, se ejecutan el resto de algoritmos para poder generar los elementos de sujeción y los paneles a cortar.

Dentro de Revit se utiliza la herramienta "Tag All" para colocar el código de cada elemento en la vista activa en ese momento, posteriormente se exportan las vistas que contienen a los elementos de sujeción y los paneles, en las propiedades de exportación se establece que la capa de modelos genéricos se represente con color rojo, estos son los elementos a cortarse, mientras que en la capa de anotación de modelos genéricos se coloca un color azul, este color representa grabado, finalmente





se exportan los resultados en formato \*.dwg, y se envía al centro de corte láser para la manufactura de las piezas.

### 3.3.2. VERIFICACIÓN DE LAS PIEZAS PARA EL ENSAMBLAJE

Una vez cortadas las piezas se verificó que todas las piezas se encuentren en estado adecuado para continuar con la elaboración del prototipo.

Se pudo notar que las piezas de MDF, tras ser cortadas con el láser presentan una coloración oscura excesiva en los bordes cortados, y en menor grado en las caras posteriores a la acción del láser, donde aparecían manchas más de manera puntual, al parecer estos fueron puntos donde el láser comenzó la acción de corte de la pieza.

Se procedió a limpiar los bordes y las caras manchadas con un paño húmedo, esto solamente por razones estéticas, mediante este proceso se pudo retirar parcialmente las manchas que dejó el láser, pero es evidente que las caras que quedarían expuestas a la vista del observador requieren de un tratamiento en este punto.

Se ha optado por la colocación de láminas de vinilo adhesivo, se ha intentado buscar un color opaco y oscuro, que sobre todo cree un contraste interesante con fuentes de luz que puedan ubicarse en la parte posterior de los paneles, al mismo tiempo

que presente un acabado mate. Las láminas utilizadas son de color plateado con un brillo muy bajo en su superficie.

Se retira el papel protector de la película adhesiva y se disponen los paneles de un mismo componente con su cara posterior directamente sobre el adhesivo, dejando así la cara codificada libre y a la vista, se unen los paneles entre sí mediante las caras laterales y en sentido anti horario se dispone a, b y c.

Una vez colocados los adhesivos sobre las caras posteriores de los paneles, los mismos quedan unidos con su panel sucesivo dejando solamente un lateral abierto, para sellar este lateral se sostiene la pieza y se doblan hacia el interior los paneles hasta que los laterales faltantes se junten, al hacerlo se crea una especie de tetraedro, que, en el grado de la abertura definida por la incidencia solar, se encuentra más o menos completo.

Una vez colocados los adhesivos en todas las piezas se coloca goma blanca de carpintero en las partes donde se juntan los paneles y del lado opuesto a donde se encuentra el vinilo, al pegar dos superficies iguales, madera con madera, la junta resultante es muy fuerte, y la pieza en sí tiene la forma en tres dimensiones más segura estructuralmente hablando es decir un tetraedro, que no permite deformaciones por estar conformado por triángulos en cada cara.

### 3.3.3. ENSAMBLAJE DEL PROTOTIPO

Una vez conformadas todas las piezas, es el momento de ensamblar el prototipo, para esto se ocupan las piezas cortadas en acrílico, estas se encuentran codificadas de tal modo que indican los paneles que actúan en la pieza, por ejemplo, el panel "a" de la pieza "4" con el panel "c" de la pieza "6", se expresa de la siguiente manera:

4a - 6c

Existen dos piezas en acrílico por cada dos paneles que se encuentran, tanto las piezas de acrílico como los paneles poseen unos cortes que encajan entre sí, fijando de manera que evite el desplazamiento el panel con la pieza acrílica, se realiza la operación de unir todos los paneles con caras continuas para la culminación del prototipo, para facilitar el ensamblaje se puede extraer el listado de códigos de cada pieza acrílica directamente de Revit, para verificar el orden de ensamblaje de los paneles.

Una vez colocados los paneles en su sitio se coloca un pegante especial que permita la fijación entre los elementos de MDF y acrílico para añadirle resistencia y estabilidad al elemento. El resultado final de todo este proceso es una superficie irregular, con perforaciones y paneles de base triangular que posee un área de 2.16 m<sup>2</sup> y dimensiones de 2.06 m x 1.36 m.





### 3.3.4. PROCESO DE ELABORACIÓN DE LA MAQUETA

Una vez elaborado el diseño de la superficie, y habiendo definido las piezas, se procede a la elaboración de la maqueta, para su construcción se utilizaron:

- Tablero de MDF de 3 mm de espesor.
- Acrílico transparente de 3 mm de espesor.
- Pegamento especializado para carpintería Elmer's Wood Glue.
- Pegamento especializado para superficies múltiples Elmer's ProBond Advanced.
- Cinta adhesiva industrial multiusos.
- Tijeras.
- Cuchillas.

En etapas anteriores de prototipado y experimentación se utilizaron los siguientes materiales:

- Cola blanca.
- Bisagras.
- Tornillos y tuercas.
- Tela de tapizar.

- Grapas de uso industrial.
- Pegamento especializado para construcción Bison MontageKit Super Strength.

La mayoría de las opciones anteriores fueron desechadas por agregar mucho peso al conjunto, y por ser poco prácticas en el proceso de ensamblaje.

#### 3.3.4.A. CORTADO DE TABLEROS DE MDF

Los paneles de MDF se cortaron en dimensiones de 1 m x 1 m y un espesor de 3 mm para aligerar el peso de la estructura, la máquina utilizada para cortar los módulos fue una sierra vertical para tableros de madera, estas máquinas tienen una profundidad de corte hasta de 75 mm.

#### 3.3.4.B. CORTADO DE LAS PIEZAS DE LA MAQUETA A LÁSER

Una vez cortados los módulos de 1 m x 1 m, estos son llevados a una máquina de grabado y corte a láser para obtener los paneles.

El concepto de corte a láser se orienta en la necesidad absoluta de precisión y exactitud de ángulos y medidas de los paneles como también en la necesidad de mantener la codificación de las piezas para su ensamblaje.



Imagen 3.34. Máquina de corte vertical para madera - Fotografía (Elaboración propia, 2016)

#### 3.3.4.C. RECUBRIMIENTO DE LAS PIEZAS CON ADHESIVO PLÁSTICO

Después de cortadas las piezas codificadas de MDF son recubiertas con un adhesivo plástico, en este caso se ha escogido un color plateado de bajo brillo para resaltar con la luz que ingrese por las perforaciones de los módulos.

#### 3.3.4.D. ARMADO DE PRISMAS CON LOS PANELES DE MDF

Antes de realizar el ensamblaje final de la maqueta se experimentó varias alternativas para unir las piezas que forman los prismas entre sí.



Con bisagra tipo libro en acero inoxidable de 1" x 1", estas piezas añadieron mucho peso a cada módulo.

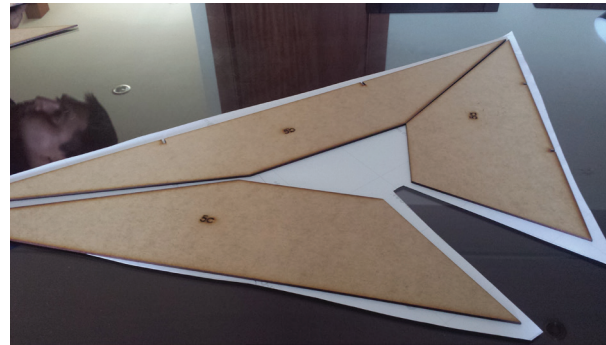
Con grapas para engrapadoras industriales y con pegamento especializado para construcción (Bison MontageKit Super Strength).

Después de experimentar con varias alternativas de unión para los paneles que forman cada prisma se descartaron estas posibilidades de unión con bisagras, grapas y macilla industrial debido a que le sumaban peso a la maqueta y retardaba el proceso de armado.

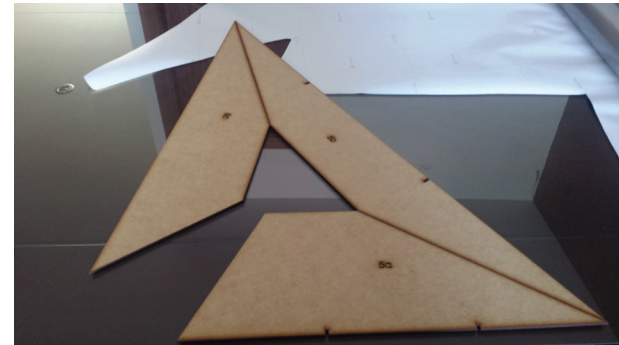
Se optó por utilizar cinta adhesiva y pegamento especializado para carpintería (Elmer's Wood Glue) como la forma más adecuada dada la resistencia del pegamento y agilidad de ensamblaje.



**Imagen 3.35.** Piezas que conforman el módulo 5 - Fotografía (Elaboración propia, 2016)



**Imagen 3.37.** Lámina adhesiva para el módulo 5 - Fotografía (Elaboración propia, 2016)



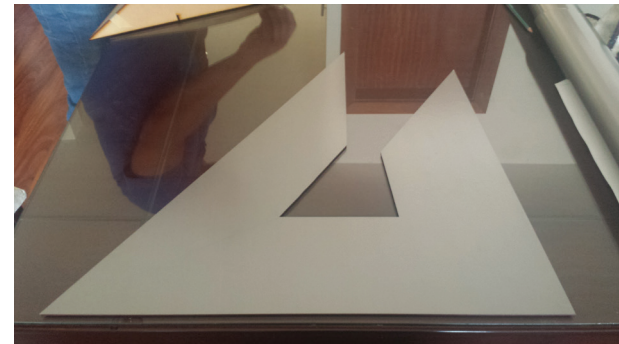
**Imagen 3.39.** Módulo 5 previo a colocar el adhesivo plástico - Fotografía (Elaboración propia, 2016)



**Imagen 3.36.** Módulos 7, 8, 9 y 11, previos a su ensamblaje - Fotografía (Elaboración propia, 2016)



**Imagen 3.38.** Lámina de adhesivo plástico - Fotografía (Elaboración propia, 2016)



**Imagen 3.40.** Módulo tras haber colocado el adhesivo plástico - Fotografía (Elaboración propia, 2016)

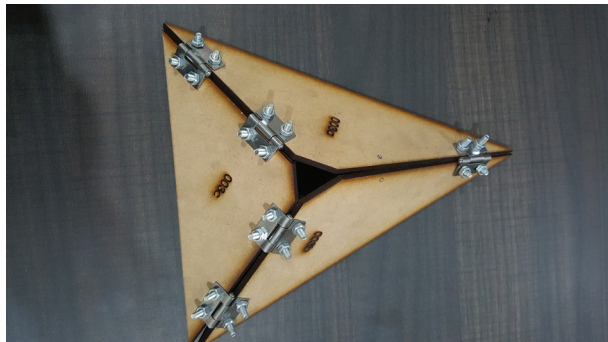


### 3.3.4.E. COLOCACIÓN DE ELEMENTOS DE SUJECCIÓN

Para unir los paneles que forman los prismas se utilizaron piezas de acrílico transparente de 3 mm de espesor cortadas a láser, las mismas que fueron previamente diseñadas en el software Dynamo, todas las piezas de acrílico y MDF están codificadas de



**Imagen 3.41.** Módulo 5 tras haber colocado la lámina de adhesivo plástico - Fotografía (Elaboración propia, 2016)

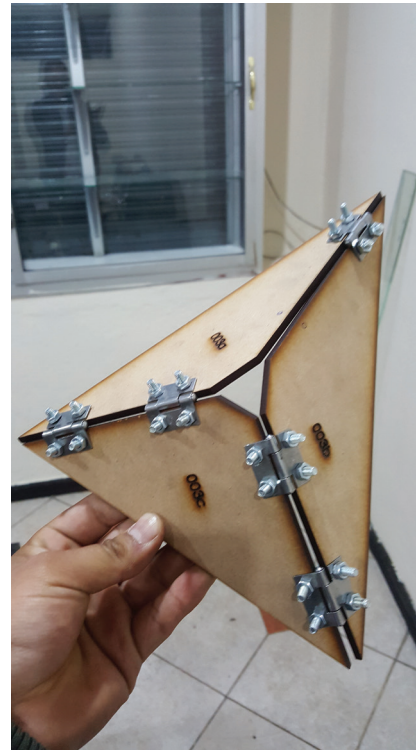


**Imagen 3.42.** Módulo 3, prototipado, experimentación con uniones con bisagras - Fotografía (Elaboración propia, 2016)

acuerdo al diseño para una adecuada forma de ensamblaje.

### 3.3.4.F. ENSAMBLAJE DE UNA MUESTRA A ESCALA 1:1

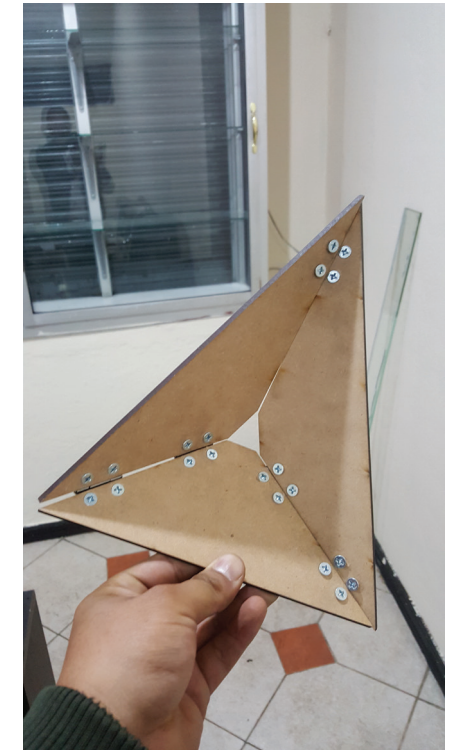
Una vez colocadas todas las piezas de sujeción se procedió a juntar todos los paneles, dando como resultado una muestra de la superficie total que se diseñó.



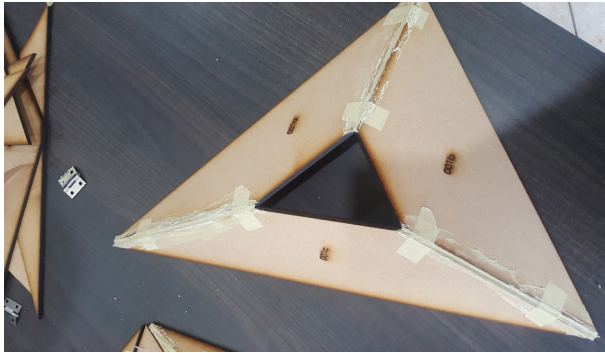
**Imagen 3.43.** Módulo 3, experimentación con uniones con bisagras - Fotografía (Elaboración propia, 2016)

### 3.3.5. OBSERVACIONES

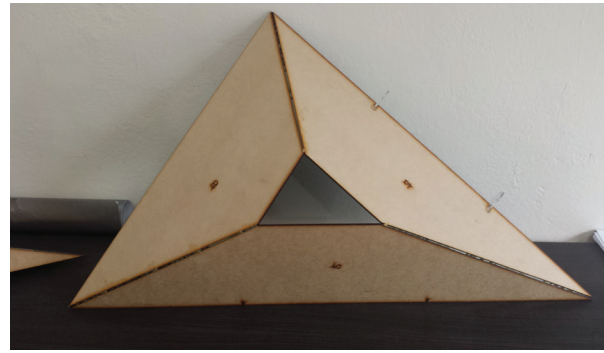
Si bien el elemento resultante es lo suficientemente liviano para ser levantado por una persona sin mayor esfuerzo, no se han corrido simulaciones estructurales ni se ha hecho algún tipo de análisis para demostrar sus capacidades estructurales para soportarse a sí mismo.



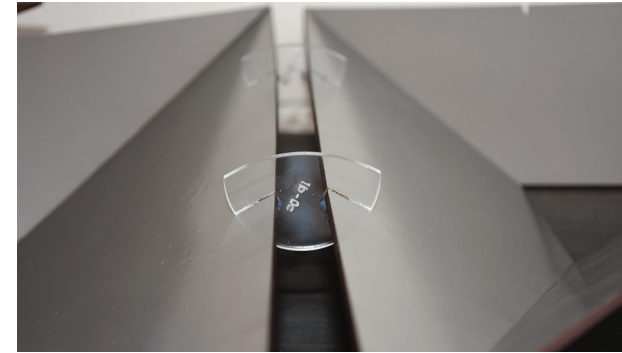
**Imagen 3.44.** Prototipado, experimentación con uniones con bisagras - Fotografía (Elaboración propia, 2016)



**Imagen 3.45.** Módulo 1, experimentación con pega de construcción - Fotografía (Elaboración propia, 2016)



**Imagen 3.47.** Módulo 4 con los elementos de sujeción acrílicos - Fotografía (Elaboración propia, 2016)



**Imagen 3.49.** Detalle del elemento de sujeción entre los paneles 0c y 1b de los módulos 0 y 1 - Fotografía (Elaboración propia, 2016)



**Imagen 3.46.** Módulos 0 y 1, etapa de prototipado - Fotografía (Elaboración propia, 2016)

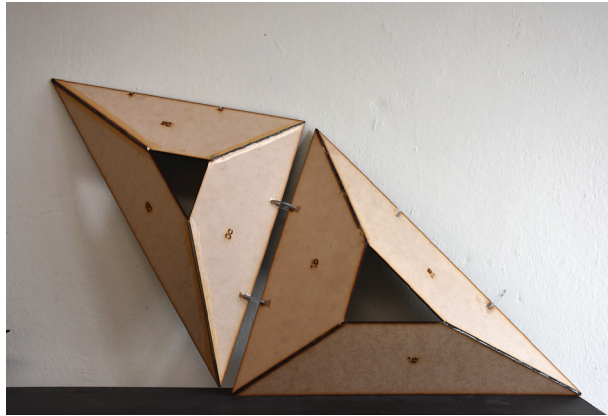


**Imagen 3.48.** Módulo 4 listo para ensamblarse con otros módulos - Fotografía (Elaboración propia, 2016)



**Imagen 3.50.** Elementos de sujeción entre módulos - Fotografía (Elaboración propia, 2016)





**Imagen 3.51.** Módulos 0 y 1 ensamblados entre sí - Fotografía (Elaboración propia, 2016)



**Imagen 3.53.** Módulos 2, 3, 7, 8 y 9 ensamblados, parte posterior - Fotografía (Elaboración propia, 2016)

Al tratarse de un elemento que funciona a manera de recubrimiento se decide que la manera de sujeción de los elementos debe ser directamente a elementos estructurales del espacio donde se colocan, es decir vigas y columnas que se encuentren vistas, mediante cables tensores que se ubiquen de manera que se distribuya acordemente el peso del elemento en los elementos que lo soportan.

Durante el ensamblaje de los módulos, se encontró como problema el peso de los paneles sobre los elementos de acrílico, los mismos que no fueron capaces de soportar el peso de los elementos una vez unidos, por lo tanto, se ha decidido reemplazar el material de estas piezas por acero, de esta manera esperando solventar los problemas que se encontraron con las piezas de acrílico.



**Imagen 3.52.** Módulos 2, 3, 7, 8 y 9 ensamblados, parte frontal - Fotografía (Elaboración propia, 2016)





## CAPÍTULO IV

### 4. CONCLUSIONES Y RECOMENDACIONES

#### 4.1. CONCLUSIONES

Durante la realización del proyecto se pudo establecer que el diseño paramétrico y su plataforma principal, los softwares de programación visual, poseen un gran potencial, tanto como herramientas generadoras de diseño y elementos, como herramientas de ayuda y automatización de procesos. Al utilizar la programación visual y sus productos, los algoritmos, se pueden realizar tareas tediosas que normalmente le tomaría a un usuario mucho tiempo realizar manualmente, en tan solo unos pocos segundos o minutos, dependiendo de la cantidad de información que se esté procesando, mediante la utilización de algoritmos.

La capacidad de los softwares de programación visual hace posible abrir la mente e imaginación a otras posibilidades y alternativas de diseño, que con métodos tradicionales ni siquiera serían consideradas, bien sea por la complejidad en su diseño o por la dificultad de poner en obra lo diseñado. Se pueden explorar nuevas formas y relaciones entre los elementos que conforman el diseño, teniendo en cuenta siempre que, a pesar de que los resultados parecen aleatorios, todo está dominado y regido por una base matemática y geométrica muy estricta

que se establece por el arquitecto desde el momento mismo que se comienza a diseñar.

Se ha observado de manera directa la capacidad de la tecnología para reducir tiempos y costos de producción, en este caso específico de los paneles para la conformación del diseño, que si se hubiera hecho de manera manual hubiera llevado más tiempo, en el caso de que hubiera sido siquiera posible, los avances tecnológicos son impresionantes, a tal punto que lo que hoy puede parecer imposible, dentro de unos años puede ser una práctica normal y hasta común.

La combinación y trabajo conjunto de herramientas CAD y CAM hace que los tiempos de trabajo para desarrollar un proyecto se disminuyan considerablemente, si a esto se incorpora el software BIM, se tiene un conjunto de herramientas de posibilidades casi ilimitadas gracias a la gran capacidad de almacenar información, procesarla y generar resultados de manera casi automatizada.

A pesar de que la tecnología para la producción mediante máquinas CNC se ha expandido en gran medida a nivel mundial, en nuestro entorno esto se ha enfocado principalmente a la industria, por lo que resulta costosa la producción de este tipo de elementos, siendo la tecnología láser la que mejores resultados

ofreció al hacer una comparación costo-tiempo.

Los resultados del desarrollo del proyecto fueron muy satisfactorios, habiendo culminado con un diseño paramétrico, al cual sin mayor intervención del usuario se le puede extraer la información necesaria para su fabricación, un listado de códigos para cada pieza y un ensamblaje ordenado, sin embargo, hubiera sido más representativo poder incluir resultados de cálculos estructurales dentro de la definición del algoritmo.

#### 4.2. RECOMENDACIONES

- La publicación del presente trabajo académico para que sea fuente de investigación sobre el tema.
- Promover la incorporación de este tipo de herramientas en el plan curricular de la carrera de Arquitectura de la Universidad de Cuenca.
- Continuar la investigación que se presenta en este trabajo para la incorporación estudios estructurales básicos en formas complejas mediante la aplicación de algoritmos.
- Establecer los programas BIM como estándar en la creación de proyectos académicos para facilitar la incorporación de la tecnología CAD/CAM en los mismos.

## BIBLIOGRAFÍA Y REFERENCIAS

- A-cero Blog.** (21 de Julio de 2015). *MUSEO AMIR DE ARTE DE TEL AVIV, POR PRESTON SCOTT COHEN*. Obtenido de <http://blog.a-cero.com/museo-amir-de-arte-de-tel-aviv-por-preston-scott-cohen/>
- Acha, J.** (2001). *Introducción a la teoria de los diseños*. Mexico: Trillas 3° edición.
- Adelto.** (20 de Noviembre de 2013). *CONSTRUCTION OF HEYDAR ALIYEV CENTER BY ZAHA HADID (PART 1)*. Obtenido de <http://www.adelto.co.uk/construction-of-heydar-aliyev-center-by-zaha-hadid/>
- AIACC.** (Junio de 2012). *Parametric design a brief history*. Obtenido de [www.aiacc.org/2012/06/25/parametric-design-a-brief-history/](http://www.aiacc.org/2012/06/25/parametric-design-a-brief-history/)
- Alvarez, R., & Roces, J.** (2005). *Introducción al diseño paramétrico*. Oviedo: Textos Universitarios Ediuno.
- Aminta Online.** (27 de Septiembre de 2010). *FASHION + ARCHITECTURE*. Obtenido de <http://www.amintaonline.com/fashion-architecture/>
- ArchDaily.** (13 de Enero de 2015). *Estudiantes mexicanos fabrican pabellón paramétrico: DIGFABMTY 1.0*. Obtenido de <http://www.archdaily.mx/mx/760111/digfabmty>
- Architizer.** (09 de Diciembre de 2016). *Archi-Quiz: Can You Identify These 17 Buildings Solely From Their Architectural Models?* Obtenido de <http://architizer.com/blog/archi-quiz-architectural-models/>
- Assael, D.** (2006). *BIM - Adios al CAD*. Obtenido de [plataformaarquitectura.cl](http://plataformaarquitectura.cl)

- Aurenhammer, F., & Klein, R.** (s.f.). Voronoi Diagrams. Obtenido de <http://www.pi6.fernuni-hagen.de/downloads/publ/tr198.pdf>
- Autodesk.** (2016). *Autodesk Knowledge Network*. Obtenido de <https://knowledge.autodesk.com/support>
- Autodesk.** (2016). *Revit LT*. Obtenido de <http://www.autodesk.es/products/revit-lt/overview>
- Averill, B., & Eldredge, P.** (2016). *General Chemistry: Principles, Patterns, and Applications*, v. 1.0 (2 Volume Set). Obtenido de [http://catalog.flatworldknowledge.com/bookhub/4309?e=averill\\_1.0-ch06\\_s01](http://catalog.flatworldknowledge.com/bookhub/4309?e=averill_1.0-ch06_s01)
- Benedetti, C.** (2012). *Influencia de la Tecnología en la Morfología de la Arquitectura*. Argentina.
- Benjamin, D., Bogosian, B., Carpo, M., Davis, D., Eisenman, P., Llaguno, M., . . . Wigley, M.** (17 de Mayo de 2014). *What Is Parametricism?*
- BobCAD-CAM.** (31 de Enero de 2014). *CAD-CAM-How CNC Manufacturing Technology is Helping Shape The World*. Obtenido de <http://bobcad.com/cad-cam-how-cnc-manufacturing-technology-is-helping-shape-the-world/>
- Bonsai Menorca.** (2015). *El Camino del agua*. Obtenido de [http://www.bonsaimenorca.com/articulos/articulos-tecnicos/el\\_camino\\_del\\_agua/](http://www.bonsaimenorca.com/articulos/articulos-tecnicos/el_camino_del_agua/)
- Browlee, D., De Long, D., & Kahn, L.** (1998). *En el reino de la arquitectura*. Barcelona: Ed. GG.

- Ciencia y Salud.** (06 de Agosto de 2012). *La estructura del agua*. Obtenido de <http://cienciaysalud.laverdad.es/la-alimentacion/la-nutricion-ciencia/la-estructura-agua-article.html>
- CMZ.** (s.f.). Tornos CNC. Obtenido de <http://www.cmz.com/tornos-cnc.php>
- Coutinho, R.** (Junio de 2013). *DESIGNA - A Shape Grammar Interpreter*.
- CutlaserCut.** (2015). *Materials*. Obtenido de [http://www.cutlaserCut.com/laser-cutting-materials-laser-engraving-materials?technique=cut\\_engrave#other-plastics](http://www.cutlaserCut.com/laser-cutting-materials-laser-engraving-materials?technique=cut_engrave#other-plastics)
- DailyTonic.** (10 de Septiembre de 2012). *Pop-up stars: temporary contemporary architecture*. Obtenido de <http://www.dailytonic.com/pop-up-stars-temporary-contemporary-architecture/>
- Deconstructivismo.** (29 de Enero de 2015). *Zaha Hadid .. Primer mujer en ganar el premio Pritzke*. Obtenido de <http://arquideconstructivismo.blogspot.com/2015/01/zaha-hadid-primer-mujer-en-ganar-el.html>
- DirectINDUSTRY.** (2016). *Centro de mecanizado CNC / 5 ejes / universal / para aluminio*. Obtenido de <http://www.directindustry.es/prod/mandelli/product-11944-460871.html>
- DirectINDUSTRY.** (2016). *CNC cutting machine / water-jet / 5-axis*. Obtenido de <http://www.directindustry.com/prod/knuth-machine-tools/product-27150-446261.html>
- DirectINDUSTRY.** (2016). *Fresadora CNC de engranajes mediante fresa madre*. Obtenido de [http://www.directindustry.es/prod/emag-gmbh-co-kg/product-15447-381779.html#product-item\\_424176](http://www.directindustry.es/prod/emag-gmbh-co-kg/product-15447-381779.html#product-item_424176)



- DirectINDUSTRY.** (2016). *TAYOR CNC fiber laser cutting machine for metal cutting*. Obtenido de <http://trends.directindustry.es/shanghai-taylor-heavy-industry-group-co-ltd/project-155580-115345.html>
- Dynamo Primer.** (2015). *Dynamo, Visual Programming*. Obtenido de <http://dynamoprimer.com/>
- Elaboración propia.** (2016).
- Equipo Arquidesign.** (29 de Abril de 2011). *OBRA ELEGIDA: "GUANGZHOU OPERA HOUSE"*. Obtenido de <http://arquidesinginc.blogspot.com/2011/04/obra-elegida-guangzhou-opera-house.html>
- Espacio Arquitectura.** (09 de Noviembre de 2013). *Diseño paramétrico, geometría avanzada*. Obtenido de <http://linearq.blogspot.com/2013/11/disenio-parametrico-geometria-avanzada.html>
- Fabricantes BIM.** (2016). *Desarrollo de catálogo BIM*. Obtenido de <http://fabricantes.bimetica.com/desarrollo-de-catalogo-bim/>
- Flores, J.** (2013). *Abstracción y estrategias en el diseño*. Mexico.
- Fuentes, A.** (22 de Noviembre de 2016). *BLOBITECTURE' AND BLOBISM; AND PARAMETRICISM*.
- García , C., & García, I.** (2010). *El diseño paramétrico y la geometría de las formas arquitectónicas*. Obtenido de [oa.upm.es](http://oa.upm.es/): <http://oa.upm.es/>
- GARQUITECTOS.** (15 de Febrero de 2016). *DYNAMO – LEVANTAR UN PARCELARIO*. Obtenido de <http://www.garquitectos.es/2016/02/15/dynamo-levantar-un-parcelario/>

- GeometricMind.** (21 de Enero de 2011). *Generative Algorithm 000*. Obtenido de <https://geometricmind.net/tag/grasshopper-2/page/6/>
- GREG LYNN FORM.** (s.f.). Greg Lynn FORM. Obtenido de BIO: <http://glform.com/bio-2/>
- Guillen, P.** (s.f.). Envoltura convexa (Convex Hull). Obtenido de [http://pier.guillen.com.mx/algorithms/07-geometricos/07.7-envoltura\\_convexa.htm](http://pier.guillen.com.mx/algorithms/07-geometricos/07.7-envoltura_convexa.htm)
- Gutierrez, M., Pérez, J., Olmo, J., & Sánchez, J.** (2012). *Incorporación del diseño y fabricación digital a la arquitectura*. Sevilla.
- Hildebrandt.** (2015). *Arquitectura*. Que es Revit y para que sirve el modelado BIM? Obtenido de <http://www.hildebrandt.cl/>
- Huang, A.** (2014). *Dynamo: Visual Programming for Revit, Optimizing Construction with BIM*. New York.
- Iwamoto, L.** (2009). *Digital Fabrications*. . New York: 1º Edición. Princeton Architectural Press.
- JPG tools & tools.** (2016). *Taladros*. Obtenido de <http://www.maquinariaindustrialcali.com/maquinaria-industrial/taladros-industriales-en-cali-armenia-pereira-manizales-popayan/>
- Khabazi, Z.** (2012). *Generative Algorithms*. Using Grasshopper.
- Kron, Z.** (29 de Agosto de 2016). *A High Level Introduction to T-Splines in Dynamo*. Obtenido de <http://dynamobim.org/a-high-level-introduction-to-t-splines-in-dynamo/>
- Landa, M.** (2001). *Deleuze y el uso del algoritmo genético en arquitectura*. Actar.

- LC&LP.** (04 de Abril de 2016). *Muere la arquitecta Zaha Hadid y la recordamos con sus mejores obras*. Obtenido de <http://lcylp.com/es/2016/04/04/muere-la-arquitecta-zaha-hadid-y-la-recordamos-con-sus-mejores-obras/>
- Lee, G., & Eastman, C.** (2006). *Specifying parametric building object behavior (BOB) for a building information modeling system*. Automation in Construction.
- Lopes, J., Paio, A., & Sousa, J.** (2014). *Rethinking Comprehensive Design: Speculative Counterculture, Proceedings of the 19th International Conference on Computer-Aided Architectural Design Research in Asia CAADRIA*. PARAMETRIC URBAN MODELS BASED ON FREI OTTO'S GENERATIVE FORM-FINDING PROCESSES. Hong Kong, China.
- Makert, R., & Alves, G.** (28 de Noviembre de 2016). *Periodica Polytechnica Architecture*. Between Designer and Design: Parametric Design and Prototyping Considerations on Gaudí's Sagrada Familia. Campo Grande, Brasil.
- MBWAREZ.** (18 de Febrero de 2011). *Tessellation in Grasshopper (Beijing Watercube revisited)*. Obtenido de [http://mbwarez.blogspot.com/2011\\_02\\_13\\_archive.html](http://mbwarez.blogspot.com/2011_02_13_archive.html)
- Molinare, A.** (2011). *Plataforma arquitectura.cl*. Obtenido de Diseño paramétrico: <http://www.plataformaarquitectura.cl/>
- Moreno, O.** (Octubre de 2012). *Sistema CAD / CAM*. Obtenido de Dibujos en 2D y 3D: [prezi.com](http://prezi.com)
- Mota, M.** (Enero de 2014). *Concepto Arquitectónico*. Obtenido de [prezi.com](http://prezi.com)
- Navarrete, S.** (Septiembre de 2014). *Diseño paramétrico*. El gran desafío del siglo XXI. Obtenido de [http://fido.palermo.edu/servicios\\_dyc/publicacionesdc/vista\\_detalle\\_articulo.php?id\\_articulo=9406&id\\_libro=470](http://fido.palermo.edu/servicios_dyc/publicacionesdc/vista_detalle_articulo.php?id_articulo=9406&id_libro=470)

- Paiz, C.** (Octubre de 2011). *Algoritmos en la Arquitectura*. Obtenido de <http://mrmannoticias.blogspot.com/>
- Parametric Fabrication.** (29 de Noviembre de 2011). *Plywood : Fabrication and Assembly*. Obtenido de <https://11arch461.wordpress.com/2011/11/29/plywood-fabrication-and-assembly/>
- Parametric Wood.** (02 de Febrero de 2011). *Relaxing Rhino Surfaces*. Obtenido de <https://parametricwood2011.wordpress.com/2011/02/02/393/>
- Perez, M.** (2013). *Nuevas formas arquitectónicas de la arquitectura paramétrica*. Obtenido de <http://blogthinkbig.com/>
- Peterson, S.** (1998). *COMPUTING CONSTRAINED DELAUNAY TRIANGULATIONS*. Obtenido de [http://www.geom.uiuc.edu/~samuelp/del\\_project.html](http://www.geom.uiuc.edu/~samuelp/del_project.html)
- Plataforma Arquitectura.** (12 de Marzo de 2015). *Pritzker 2015: Frei Otto y la importancia de la experimentación en la arquitectura*. Obtenido de <http://www.plataformaarquitectura.cl/cl/763636/pritzker-2015-frei-otto-y-la-importancia-de-la-experimentacion-en-la-arquitectura>
- Quiminet.** (2012). *El sistema CAD/CAM*. Obtenido de <http://www.quiminet.com/>
- QuimiTube.com.** (s.f.). Enlaces de hidrógeno y fuerzas de van der Waals. Obtenido de <http://www.quimitube.com/videos/enlaces-de-hidrogeno-y-fuerzas-de-van-der-waals>
- Revista PLOT.** (06 de Agosto de 2012). *ENTREVISTA A GREG LYNN*. Obtenido de <http://www.revistaplot.com/en/entrevista-a-greg-lynn/>
- Rodriguez, H.** (2014). *Documento de Arquitectura de Software*. Ecuador.

- Schumacher, P.** (2008). *Parametricism Manifesto*. Obtenido de [www.patrickschumacher.com](http://www.patrickschumacher.com)
- Schumacher, P.** (26 de Junio de 2009). *Parametricism. A New Global Style for Architecture and Urban Design*.
- SHL GROUP.** (2016). *CNC MACHINING*. Obtenido de <http://www.shl-group.com/capabilities/cnc-machining/>
- Simply Rhino Rhino3D Channel.** (12 de Julio de 2012). *La visualización de arquitectura en Rhino V5*. Obtenido de <https://www.youtube.com/watch?v=bkrh8QAKgjc>
- SolidSmack.** (26 de Octubre de 2009). *AutoDesk Inventor Fusion Adds Thick, Beefy Change Management to Manage... Your... Change*. Obtenido de <http://www.solidsmack.com/cad-design-news/autodesk-inventor-fusion-adds-change-manager-workflow/>
- Soriano, A.** (08 de Enero de 2013). *DIAGRAMAS DE VORONOI*. Obtenido de <https://www.alsomar.com/2013/01/diagramas-de-voronoi/>
- Tan, A.** (11 de Julio de 2014). *Architecture e-portfolio*. Obtenido de <http://tanhongloong-e-portfolio.blogspot.com/2014/07/shape-grammar-arc3153-arc3152-module.html>
- Tedeschi, A.** (2011). *Intervista a David Rutten*. Mix Experience Tools 1. Napoles.
- Visualarq.** (2015). <http://www.visualarq.com/>. Obtenido de Que es Rhino?: <http://www.visualarq.com/>
- Wassim , J.** (2013). *Parametric Design for Architecture*. Londres: Laurence King.
- Wiens, D.** (28 de Diciembre de 2013). *PaperJS Voronoi Diagrams*. Obtenido de <https://darrenwiens.wordpress.com/2013/12/28/paperjs-voronoi-diagrams/>



- Wikimedia Commons.** (13 de Abril de 2007). *File:Delaunay Triangulation (50 Points).svg*. Obtenido de [https://commons.wikimedia.org/wiki/File:Delaunay\\_Triangulation\\_\(50\\_Points\).svg](https://commons.wikimedia.org/wiki/File:Delaunay_Triangulation_(50_Points).svg)
- Wikipedia.** (18 de Octubre de 2016). *Dispersion (water waves)*. Obtenido de [https://en.wikipedia.org/wiki/Dispersion\\_\(water\\_waves\)](https://en.wikipedia.org/wiki/Dispersion_(water_waves))
- Wikipedia.** (20 de Noviembre de 2016). *Laser cutting*. Obtenido de [https://en.wikipedia.org/wiki/Laser\\_cutting](https://en.wikipedia.org/wiki/Laser_cutting)
- Wikipedia.** (27 de Julio de 2016). *Parametric Design*. Obtenido de [https://en.wikipedia.org/wiki/Parametric\\_design](https://en.wikipedia.org/wiki/Parametric_design)
- Woodbury, R.** (2010). *Elements of Parametric Desingn*. New York: Routledge.
- WordPress.com.** (2013). *Lenguaje de Ingenieria*. Obtenido de Sistemas CAD / CAM: [lenguajedeingenieria.files.wordpress.com](http://lenguajedeingenieria.files.wordpress.com)